A RAND NOTE

Year of Service Target Generator: Mathematical Specification

Grace M. Carter

RAND

The research reported here was sponsored by the United States Air Force under Contract F49620-86-C-0008. Further information may be obtained from the Long Range Planning and Doctrine Division, Directorate of Plans, Hq USAF.

The RAND Publication Series: The Report is the principal publication documenting and transmitting RAND's major research findings and final research results. The RAND Note reports other outputs of sponsored research for general distribution. Publications of RAND do not necessarily reflect the opinions or policies of the sponsors of RAND research.

Published 1991 by RAND
1700 Main Street, P.O. Box 2138, Santa Monica, CA 90407-2138

A RAND NOTE

N-3223/2-AF

Year of Service Target Generator: Mathematical Specification

Grace M. Carter

Prepared for the
United States Air Force

RAND

PREFACE

RAND is helping to design an Enlisted Force Management System (EFMS) for the Air Force.[1] The EFMS is a decision support system to assist managers of the enlisted force in setting and meeting force targets. The Year of Service Target Generator (YOSTG) produces desirable year of service distributions for each occupation in the enlisted force. The distributions give each occupation a level of experience consistent with the grade distribution found in authorizations. The targets are designed to help in the management of year-group programs such as bonuses and career force entry. This Note gives the detailed mathematical specification of the YOSTG. It is written for the analysts who will implement the model and for users who wish to understand details of the model's structure and operation. It is intended to be used in conjunction with a companion volume, Grace M. Carter, *Year of Service Target Generator: Conceptual Specification,* N-3223/1-AF, which provides the conceptual foundations of the model, an overview of how the model works, and examples of its output.

The work described here is part of the Enlisted Force Management Project (EFMP), a joint effort of the Air Force (through the Deputy Chief of Staff for Personnel) and The RAND Corporation. RAND's work falls within the Resource Management and System Acquisition Program of Project AIR FORCE. The EFMP is part of a larger body of work in that program concerned with the effective utilization of human resources in the Air Force.

[1] For an overview of the EFMS see Grace M. Carter, Jan M. Chaiken, Michael P. Murray, and Warren E. Walker, *Conceptual Design of an Enlisted Force Management System for the Air Force,* RAND, N-2005-AF, August 1983.

ACKNOWLEDGMENTS

An earlier version of this Note was reviewed by Major Ramon Cortes, TSgt William Scott of DPDW, and Jacob Klerman of RAND. TSgt Scott wrote code for the operational model and uncovered many problems with the specification details. I am very grateful for their corrections and suggestions for improvement. Any errors that remain are, of course, my responsibility.

CONTENTS

FIGURES

TABLES

GLOSSARY

AFSC	Air Force Specialty Code (an occupational designation)
BEM	Bonus Effects Model
CAREERS	Program for managing retraining at first term reenlistment
CJR	Career job reservation
DMI	Disaggregate Middle-term Inventory projection model
EFMS	Enlisted Force Management System
ETS	Expiration of term of service
IPM	Inventory Projection Model
NPS	Non-prior service
SRB	Selective reenlistment bonus
SSL	Self-sustaining ladder
TOE	Term of enlistment
YETS	Years to expiration of term of service
YOS	Year of service
YOSTG	Year Of Service Target Generator

I. INTRODUCTION

This Note contains the mathematical specification of the Year of Service Target Generator (YOSTG) of the Enlisted Force Management System (EFMS). The YOSTG produces desirable year of service (YOS) distributions for each occupation in the enlisted Air Force. The YOS targets are designed to meet mission needs as reflected in authorizations and to be attainable with current personnel policies.

This Note describes the model's equations and the algorithms used to solve them. It is directed toward the analysts who will implement the model and those who wish to understand the details of the model's structure and operation. A companion Note (Carter, 1991) describes the conceptual foundations of the model and provides an overview of how the model works. This Note is not meant to be read by itself. The reader should understand the material presented in the conceptual specification.

The YOSTG consists of five modules, as shown in Fig. 1. The first module performs all data preparation necessary for the model. Within this module, authorizations and inventory are grouped into self-sustaining ladders (SSLs). Files are transformed so that all other data (e.g., loss and promotion rates) on each self-sustaining ladder are in the format expected by the other modules.

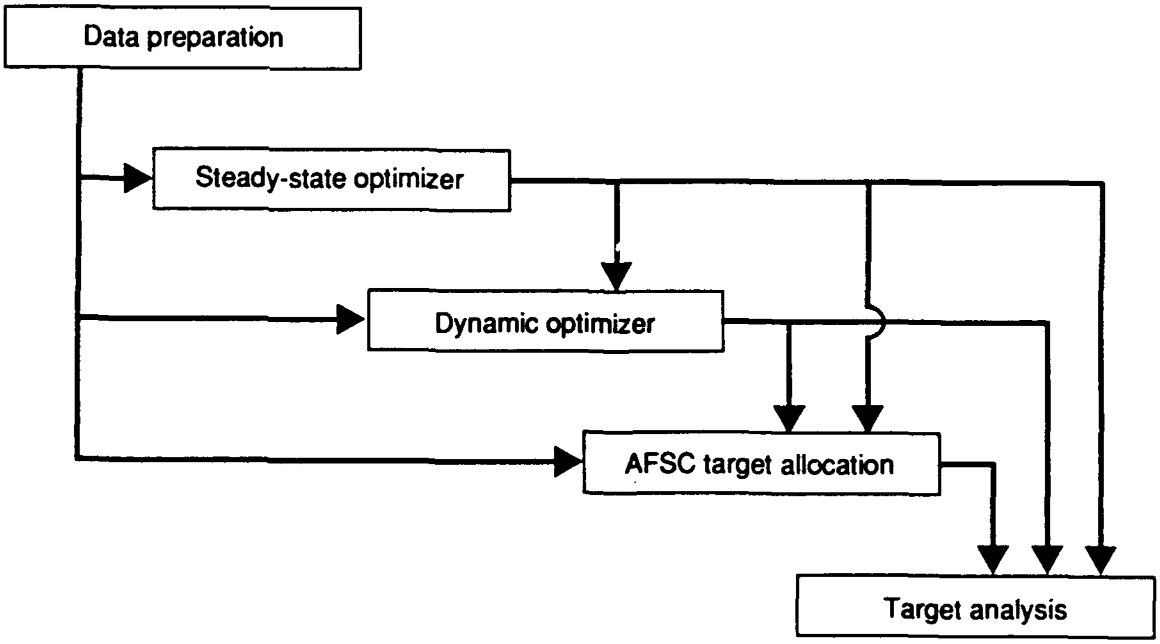

Fig. 1—Modules of the YOSTG

The second module, the steady-state optimizer, produces a steady-state target for each self-sustaining ladder. This target is part of the input to the third module of the system, the dynamic optimizer. The final step necessary to generate a target is the allocation of the targets for self-sustaining ladders to individual Air Force Specialty Codes (AFSCs). The steady-state target and the dynamic targets are allocated within the fourth module.

The model's primary output comes from the AFSC target allocation module. The output is a file giving the target inventory in each AFSC in each Inventory Projection Model (IPM) cell in each year of the planning horizon and in steady-state. This output allows calculation of the target force by any combination of YOS, grade, and category of enlistment. The fifth module facilitates examination of these targets.

The remaining sections of this Note correspond to the first four of the five modules. These specifications are complete and correspond to the operations of the prototype version, unless explicitly noted. In the prototype version, the targets are displayed only within the Bonus Effects Model (BEM), and thus the prototype contains no separate fifth module. However, as the model is used for additional purposes, as suggested in the conceptual specification Note, it will be useful to have displays designed to serve those purposes. Formulas that could be used to calculate statistics are presented. These formulas are found with the discussion of existing output from each module. Further examples of ways to summarize the data are given in Carter, 1991.

The main modules within the YOSTG retain different parts of the same data within core and, therefore, use different subscripts to access the data. For example, authorizations for SSLs are generated for each combination of ladder, grade, and planning year. The program that calculates these authorizations indexes them only by ladder and grade because different years of data are processed in separate runs. In the steady-state optimizer the same authorizations are indexed only by grade, because the data for only one ladder reside in core at any one time and only one year's worth of authorizations are needed. In the dynamic target generator the indexes are grade and planning year. To avoid confusion about the meaning of indexes, this Note uses different variable names for each index set. The connections among the different variable names are discussed in the input and output subsections.

This Note provides only mathematical specifications, not detailed specifications for the design of a computer program. Nevertheless, in many parts of Secs. III and IV, the most direct way to describe the necessary relationships was to embrace the "if-then" expression and DO loop conventions of an algorithmic programming language. I use the PL/1 form of these conventions.

II. PREPARING DATA FOR SELF-SUSTAINING LADDERS

The occupational unit used to group inventory and authorizations within the optimizing programs of the YOSTG is called a self-sustaining ladder. After the model finds the optimal target inventory for each SSL, it allocates that target among the AFSCs whose authorizations are part of the SSL.

The model uses SSLs (rather than simply using AFSCs) because each is independent, greatly increasing the efficiency of the optimizing program. The flows into capper AFSCs or into lateral entry AFSCs are determined by personnel policy. After one has accounted for these flows by grouping authorizations and inventory into SSLs, one can optimize separately for each SSL.

There is one SSL for each basic AFSC—for each AFSC that is neither a capper nor a lateral. Each self-sustaining ladder has associated with it a set of authorizations consisting of all the authorizations for the basic AFSC plus a share of all the cappers and laterals that the AFSC feeds either directly or indirectly (e.g., by feeding a lateral that feeds another lateral). So the sum of the authorizations in any grade across all SSLs equals total Aiı Force authorizations for trained personnel in the grade. One can think of the authorizations associated with an SSL as the set of all the positions that would be filled in steady state by the non-prior service (NPS) accessions who entered the force in the basic AFSC.

A full YOSTG model run begins when each year's authorizations are grouped into the set of SSLs that will be used for the run. The data needed for this process consist of (1) counts of authorizations for trained inventory in each combination of grade and AFSC for each year of the planning horizon (i.e., for T = 1,2,... PLANYEARS), (2) the capper table, and (3) the lateral-feeder table.

The next two subsections discuss the grouping process in the order in which it occurs: First, capper authorizations are grouped with their feeders; then, lateral entry AFSC authorizations are grouped with their feeders. The optimizing programs to be described in Secs. III and IV run inventory projection models for each SSL. Within the EFMS, the data needed for these Inventory Projection Models (IPMs) are stored in arrays indexed by AFSC. The third subsection describes how these data are associated with SSLs.

GROUPING CAPPER AUTHORIZATIONS

When airmen in some AFSCs are promoted to some grades, they automatically change their AFSC. The new AFSC that they enter is called a capper AFSC, and the grade at which the capper is entered is the capper grade. Currently, the only capper grades are E-9, E-8, and E-6. The number of persons in a capper AFSC depends on the number of persons in the grade below the capper grade in each feeder AFSC, the promotion rate to the capper grade, and the loss rate from the capper.

The process of grouping authorizations into SSL authorization sets allocates each E-9 capper's authorizations to all of its feeder AFSCs, including those that are themselves cappers or laterals. The allocation to each feeder is in proportion to the number of E-8 authorizations in that feeder. Then each E-8 capper's authorizations (including its share of an E-9 capper, if any) are allocated to all its feeders in proportion to the number of E-7 authorizations in each feeder. Finally, the E-6 capper authorizations are allocated analogously based on E-5 authorizations.

Because the process is repeated in the identical manner for each year's authorizations, the subscript T is omitted in the algorithm that follows. The algorithm is illustrated in Table 1, which uses data from all AFSCs from which one can eventually be promoted to AFSC 23100 to illustrate the allocation of capper authorizations to basic or lateral AFSCs. At each step, lines for the capper AFSCs whose authorizations have been used are deleted from the table, as the program never uses this information again. The allocations of authorizations are calculated and kept as fractions rather than being rounded to integers.

The algorithm is:

Step 1: Initialize the variable AUTHC(I,J) to the count of authorizations for grade J and AFSC I for all I and J.

For each AFSC I, initialize the variable CAP(I) to the AFSC that caps I if there is one and to a missing value (e.g., "xxxxxx") otherwise.

Step 2A: For each E-9 capper, IC, compute:

$$\mathrm{FEEDCN(IC)} = \sum_{I} \mathrm{AUTHC(I,8)},$$

where the sum is over all AFSCs I such that CAP(I) = IC.

Step 2B: For each I such that CAP(I) is an E-9 capper, compute

$$\mathrm{AUTHC(I,9)} = \mathrm{AUTHC(I,9)} + \mathrm{AUTHC(CAP(I),9)} \times \mathrm{AUTHC(I,8)} / \mathrm{FEEDCN(CAP(I))}.$$

Table 1

Example of Allocation of Capper Authorizations to Feeders

		Authorization Counts						
AFSC	Capper	E3	E4	E5	E6	E7	E8	E9
				Initial Position				
23100	None	0	0	0	0	0	0	27
23199	23100	0	0	0	0	0	40	0
231X0	23199	91	70	90	35	22	0	0
231X1	23199	160	161	204	96	42	0	0
231X2	23199	228	203	170	84	61	0	0
232X0	23100	51	58	69	43	22	3	0
233X0	23100	276	194	189	88	52	12	0
				After Allocating E-9 Cappers				
23199	23100	0	0	0	0	0	40	19.6
231X0	23199	91	70	90	35	22	0	0
231X1	23199	160	161	204	97	42	0	0
231X2	23199	228	203	170	84	61	0	0
232X0	23100	51	58	69	43	28	3	1.5
233X0	23100	276	194	189	88	52	12	5.9
				After Allocating E-8 Cappers (final position)				
231X0	23199	91	70	90	35	22	7.0	3.5
231X1	23199	160	161	204	97	42	13.4	6.6
231X2	23199	228	203	170	84	61	19.5	9.6
232X0	23100	51	58	69	43	28	3.0	1.5
233X0	23100	276	194	189	88	52	12.0	5.9

Step 3A: For each E-8 capper IC, compute:

$$\text{FEEDCNT(IC)} = \sum \text{AUTHC(I,7)}, \qquad \text{I}$$

where the sum is over all AFSCs I such that CAP(I) = IC.

Step 3B: For each I such that CAP(I) is an E-8 capper, compute

$$\text{AUTHC(I,9)} = \text{AUTHC(I,9)} + \text{AUTHC(CAP(I),9)} \times \text{AUTHC(I,7)} / \text{FEEDCNT(CAP(I))},$$

and

$$\text{AUTHC(I,8)} = \text{AUTHC(I,8)} + \text{AUTHC(CAP(I),8)} \times \text{AUTHC(I,7)} / \text{FEEDCNT(CAP(I))},$$

Step 4A: For each E-6 capper, IC, compute:

$$\text{FEEDCNT(IC)} = \sum_{\text{I}} \text{AUTHC(I,5)},$$

where the sum is over all AFSCs I such that CAP(I) = IC.

Step 4B: For each I such that CAP(I) is an E-6 capper, compute for each J = 6 to 9

$$AUTHC(I,J)=AUTHC(I,J)+AUTHC(CAP(I),J)\times AUTHC(I,5)/FEEDCNT(CAP(I)).$$

GROUPING AUTHORIZATIONS FOR LATERAL ENTRY AFSCs

Lateral entry AFSCs are special AFSCs that do not receive NPS accessions, so everyone entering these AFSCs has served in another AFSC. The lateral-feeder table contains one record for each lateral entry specialty. Each record lists all the AFSCs that normally feed the lateral AFSC and the percent of the entrants to the lateral specialty that will come from each feeding AFSC.

After capper authorizations have been grouped with their feeders, the process of grouping authorizations into self-sustaining ladders is completed by grouping lateral authorizations (including their share of capper authorizations) with authorizations from basic AFSCs (which also include their share of capper authorizations). The allocation of each lateral's authorizations among its feeders is determined by the proportions found in the lateral-feeder table.

As a preliminary step, the lateral-feeder table is processed recursively to replace each listed feeder that is itself either a capper or a lateral with the set of basic AFSCs that will sustain the listed feeder. The result of this preliminary step is called the "lateral-basic-feeder table."

Table 2 illustrates the creation of the lateral-basic-feeder table using a very simple lateral-feeder table and the algorithm described below. Each entry in the lateral-feeder table is shown within parentheses. In the example, AFSC 23199 contributes 30 percent of the entries to lateral specialty 100X0. Within the algorithm, the original lateral-feeder table is placed in a data structure such that, for each lateral, IL, there is a list containing feeder AFSCs, FAF(IL,INDEX),INDEX =1,2,. . . and variables X(IL,INDEX), such that X(IL,INDEX) is the fraction of entrants to lateral entry specialty IL that come from FAF(IL,INDEX). In this example, for IL = 100X0, FAF(100X0, 1) = 23199, X(100X0, 1) = 0.30, FAF(100X0, 2) = 276X0, X(100X0, 2) = 0.30, etc.

The first three steps eliminate cappers from the list of feeders in the lateral feeder table:

Step 1. For each IL, search the list of feeders to lateral IL for each occurrence of an E-9 capper. If IC = FAF(IL,INDEX) is an E-9 capper, then set XC = X(IL,INDEX) and remove

Table 2

Creation of Lateral Basic Feeder Table

Lateral AFSC	List of Feeders with Fraction of Lateral Entrants from Each Feeder
	Initial Lateral Feeder Table
100X0	(23199, 0.30); (276X0, 0.30); (995X2, 0.40)
113X0C	(426X3, 0.75); (431X2, 0.25)
732X4	(702X0, 0.75); (732X0, 0.25)
995X2	(603X0, 0.20); (732X4, 0.80)
	After Replacing Cappers with Feeders (Step 3)
100X0	(231X0, 0.053); (231X1, 0.101); (231X2, .146); (276X0, 0.30); (995X2, 0.40)
113X0C	(426X3, 0.75); (431X2, 0.25)
732X4	(702X0, 0.75); (732X0, 0.25)
995X2	(603X0, 0.20); (732X4,0.80)
	After First Iteration of Step 4
100X0	(231X0, 0.053); (231X1, 0.101); (231X2, 0.146); (276X0, 0.30); (603X0, 0.08); (732X4, 0.32)
113X0C	(426X3, 0.75); (431X2, 0.25)
732X4	(702X0, 0.75); (732X0, 0.25)
995X2	(603X0, 0.20); (702X0, 0.60); (732X0, 0.20)
	After Second Iteration of Step 4 (final position)
100X0	(231X0, 0.053); (231X1, 0.101); (231X2, 0.146); (276X0, 0.30) (603X0, 0.08); (702X0, 0.24); (732X0, .08)
113X0C	(426X3, 0.67); (431X2, 0.33)
732X4	(702X0, 0.75); (732X0, 0.25)
995X2	(603X0, 0.20); (702X0, 0.60); (732X0, 0.20)

FAF(IL,INDEX) and X(IL,INDEX) from their lists. Add an entry to the list of feeders for each I for which IC = CAP(I), and associate it with the feeder fraction:

$$XC \times AUTHC(I,8)/FEEDCNT(IC).$$

Step 2. For each IL, search the set of feeders to lateral IL for each occurrence of an E-8 capper. If IC = FAF(IL,INDEX) is an E-8 capper, then set XC = X(IL,index) and remove FAF(IL,INDEX) and X(IL,INDEX) from their lists. Add an entry to the list of feeders for each I for which IC = CAP(I) and associate it with the feeder fraction:

$$XC \times AUTHC(I,7)/FEEDCNT(IC).$$

Step 3. For each IL, search the set of feeders to lateral IL for each occurrence of an E-6 capper. If IC = FAF(IL,INDEX) is an E-6 capper, then set XC = X(IL,INDEX) and remove FAF(IL,INDEX) and X(IL,INDEX) from their lists. Add an entry to the list of feeders for each I for which IC = CAP(I), and associate it with the feeder fraction:

$$XC \times AUTHC(I,5) / FEEDCNT(IC).$$

The following step is performed repeatedly until there are no more laterals listed as feeders in the lateral feeder table.

Step 4. For each IL, search the set of feeders to lateral IL for each occurrence of a lateral. If LL = FAF(IL,INDEX) is a lateral, then set XC = X(IL,INDEX) and remove FAF(IL,INDEX) and X(IL,INDEX) from their lists. Add an entry to the list of feeders of IL for each I for which I = FAF(LL,JNDEX) for each value of JNDEX, and associate it with the feeder fraction:

$$XC \times X(LL, JNDEX).$$

This process terminated after performing either three or four iterations of Step 4 for the data in the lateral-feeder tables that existed in each of the last three fiscal years. The program should put a maximum on the number of times that Step 4 can be performed (say 10), because it is not logically necessary that the procedure terminate. The procedure would not terminate only if the lateral feeder table contained a circle so that one lateral supported another lateral, which in turn supported the first ladder (either directly or indirectly). This case will probably never arise because it would then be impossible to calculate the trained personnel requirement for the laterals using the standard methodology. Should it arise, it would be necessary for the user to exercise some judgment about how to break the circle.

After the lateral feeder table has been processed so that all listed feeders are basic AFSCs, it is a simple matter to group the authorizations for the laterals with those of the appropriate basic AFSCs. Thus the last step in the procedure is:

Step 5. For each lateral AFSC IL, and for each entry in its feeder list, and for each grade J, set:

$$\begin{aligned} AUTHC(FAF(IL,index),J) = {} & AUTHC(FAF(IL,INDEX),J) + \\ & X(IL,INDEX) \times AUTHC(IL,J). \end{aligned}$$

OUTPUT

At the conclusion of Step 5, a file is output containing one record for each basic AFSC. Each record gives the name of the basic AFSC, which is then used as the name of the SSL throughout the rest of the model. Each record also gives the authorizations associated with the SSL:

$$(AUTHC(I,3), AUTHC(I,4), \ldots AUTHC(I,9)).$$

A separate file of lateral authorizations is also output. This file contains one record for each lateral AFSC. The record contains the authorizations by grade for the lateral and its share of capper authorizations, if any; i.e., the file contains:

$$(AUTHC(IL,3), AUTHC(I,4), \ldots AUTHC(IL,9)).$$

This vector was not changed during the process of allocating lateral authorizations to SSLs nor in the process of creating the lateral-basic-feeder table. Thus it contains the same values that it had at the end of the process allocating the capper authorizations.

The processed version of the lateral feeder table in which all listed feeders are basic AFSCs is also output. This will be used to allocate the target to the AFSC level and to process other input to the optimizing model. This is called the lateral-basic-feeder table to distinguish it from the original lateral-feeder table.

The optimizing programs to be described in Secs. III and IV run Inventory Projection Models (IPMs) for each self-sustaining ladder. Within the EFMS, the data needed for these IPMs are stored in arrays indexed by AFSC. This subsection describes how to assemble these data into arrays indexed by SSL. The creation of these files completes the data preparation task.

The initial inventory associated with the self-sustaining ladder consists of all the airmen in the basic AFSC plus a share of the airmen in capper and lateral specialties. Each SSL receives exactly the same proportion of inventory in a given AFSC as it receives of authorizations in that AFSC. The calculations for cappers are directly analogous to those given for authorizations. The calculations for laterals are similarly straightforward using the lateral-basic-feeder table.

In principle, the loss rates associated with each IPM cell for each SSL should be calculated as a weighted average of the loss rates for the same cell in each AFSC that contributes authorizations to the SSL, with the weight being equal to the proportion of authorizations in that grade contributed to the SSL by the AFSC. In practice, the loss rates associated with the basic AFSC are a very good approximation to this ideal and are used in the current prototype. (In the first and second terms, almost all authorizations come from the basic AFSC; in the career and retirement-eligible cells the loss rates differ by either career field or career field group, and almost all authorizations come from the same career field.)

All other inputs to the IPM are based on the input for the basic AFSC. These include training data, career field group, and promotion tier.

III. STEADY-STATE OPTIMIZER

This section contains the specifications for the module that calculates the steady-state YOS target for each self-sustaining ladder. An overview of the algorithm includes the order in which calculations occur. Following subsections describe the input data, how flow rates are adjusted to reflect given values of the decision variables, how the steady-state solution is calculated from these flow rates, how improved values of the decision variables are calculated from derivatives of the penalty function, and the criteria for convergence. The last subsection describes the output from the module.

OVERVIEW

There is exactly one steady-state inventory that has any given set of continuation rates, promotion rates, and end strength. The promotion rates, end strength, and most of the continuation rates are given in this problem. The remaining continuation rates are those corresponding to end of term decisions by year groups that are eligible for a bonus. The middle term loss model specifies functions that relate these continuation rates and reenlistment rates to the bonus multiple offered in each zone. Thus, one may view the steady-state inventory as a function of the bonus offer in each zone, or, more precisely, as a function of the decision variables SSB(Z), for Z = 1, 2, 3 that act as bonus levels when they are within the range from 0 to 6.

The optimal steady-state inventory for each self-sustaining ladder is obtained separately. Figure 2 presents the flow for a single SSL. Each of the boxes represents a subroutine (many of which call other subroutines that will be described later). The initialization subroutine reads input data. One of these input data sets gives the initial values of SSB(Z). The first time that control enters the subroutine in the chart labeled "Adjust flow rates," the loss rates, reenlistment rates, CAREERS inflow rate, and term of enlistment proportions are adjusted to reflect these initial values. The next step finds the steady-state solution that corresponds to the current values of the flow rates. In the following step, the steepest descent method is used to find values for each of the three decision variables that will reduce the penalty function. These values are compared with SSB(Z). If they are similar the solution is deemed acceptable, the target inventory and some other data are output, and the run for this SSL ends. Otherwise, it is necessary to set SSB(Z) for each Z equal to the chosen values and return to the step that adjusts the flow rates.

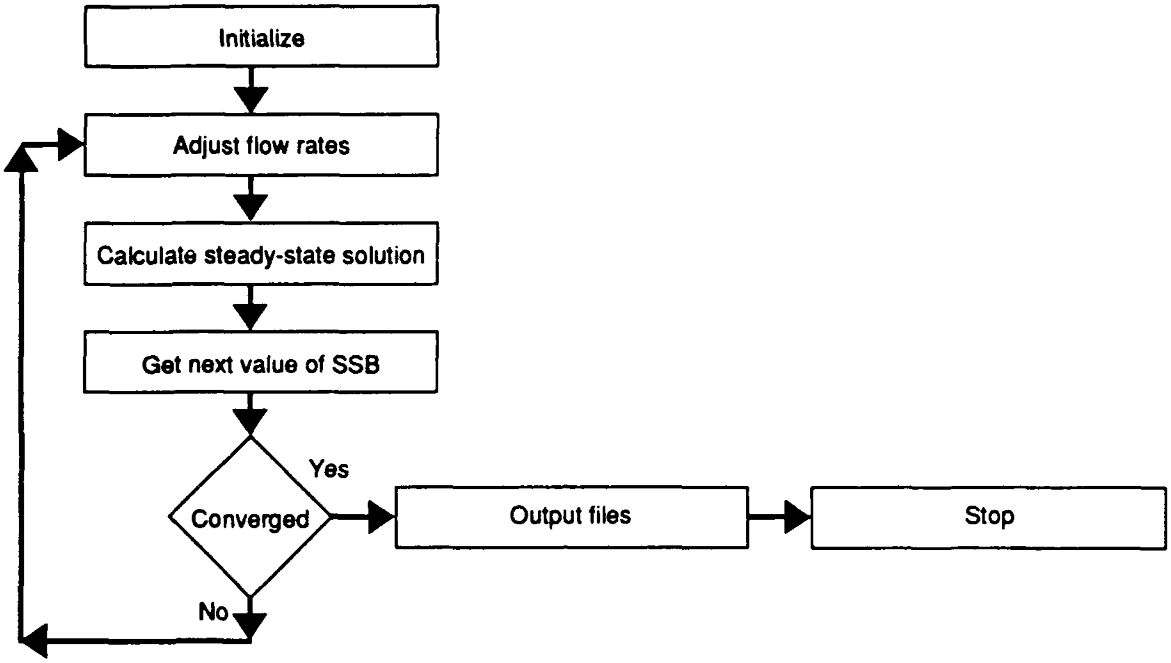

Fig. 2—Overview of steady-state model

INITIALIZATION

The initialization subroutine of the module reads in (or otherwise assembles) the data needed for the run. After the subscripts used to index the data are defined the data are discussed beginning with those that define the penalty function and control the optimization procedure, and then the data relating to flow rates.

Subscripts

Table 3 lists the subscripts that index data within the steady-state target generator. Note that there is no index pointing to which SSL is being considered. Although almost all data will be chosen specifically for a particular self-sustaining ladder, the initialization phase chooses the appropriate data and places it in arrays without an AFSC index. This simplifies describing the algorithm as well as the program itself.

The inventory within the model is counted in cells defined by combinations of grade (J in our notation), years-to-ETS (YETS), YOS, and category of enlistment (C). The Disaggregate Middle-Term IPM (DMI) uses these same cells, but it also includes dimensions for AFSC and planning year, which are not necessary here. Some combinations of these subscripts can not occur in the real world, and it may be desirable to save space and

Table 3

Subscripts for Steady-State Target Generator

Name	Range	Meaning
C	1 to 4	Category of enlistment
J	3 to 9	Grade
YETS	– 1 to 6	Years before ETS
YOS	0 to 29	Years of service
Z	1 to 3	Bonus zone
BM	0 to 6	Bonus multiple

processing time by establishing pointers giving the acceptable ranges for combinations of variables. This is done in the prototype. However, in the interests of clarity and brevity, the specifications reported here ignore this possible refinement. The phrase "DO OVER subscript" means the allowable range for the variable within the limits of the other variables.

Control Variables

The first part of Table 4 lists the variables that enter the penalty function and control the optimization procedure. In terms of these variables, the penalty function is expressed as:

$$\mathrm{SSOBJ} = \sum_{J} W(J) \times [\mathrm{SSFG}(J) - \mathrm{SSA}(J)]^2, \tag{3.1}$$

where SSFG(J) = number of persons in grade J in the steady-state force.

Promotion Rates

The third section of Table 4 lists the variables that control the flows among the model's cells. Promotion from E-3 to E-4 is on a "fully qualified basis," which means that all who meet the required standards will be promoted. The times of promotions differ for 4-year enlistees and for 6-year enlistees, hence the variable PRATE3 is dimensioned by both YOS and YETS.

Promotion out of grades E-4 and above is done at a rate that will meet planned end strength in the aggregate force. Each AFSC then receives a promotion rate for each grade that differs from the Air Force-wide select rate for that grade by an amount that depends on promotion tier. These promotion rates are stored in array SELECT.

To go from the promotion rate for a grade to a promotion rate for each cell in the model, an algorithm is used that is based on the promotion model used in the DMI of the EFMS. The promotion model is based on unpublished work by Captain Jan D. Eakle-

Table 4

Input Variables for the Steady-State Target Generator

Name	Subscripts	Meaning
		Penalty Function
SSA	J	Authorizations by grade for the particular SSL and the farthest planning year available (AUTHC of Sec. II)
W	J	Weight for grade in the penalty function
		Optimization Control Variables
STEPTOL	—	Tolerance factor in the convergence test
MAXLOOP	—	Maximum number of iterations allowed in the convergence test
SSB	Z	Decision variable (similar to bonus multiple for zone)
BBOT	Z	Lower bound for SSB
BTOP	Z	Upper bound for SSB
		Flow Rates for Inventory Projection
BETA	Z	Bonus coefficient in loss model equation
BETAE	Z	Bonus coefficient in extend-given-stay model equation
BETALB	Z	Coefficient on bonus value received at previous reenlistment in ETS loss model equation
BONPAY	Z	Average basic pay for reenlistees in zone Z
BONCAP	—	Maximum amount of bonus offer (in dollars)
CAREERS	BM	Flow into the AFSC in the CAREERS program given each bonus multiple from the Bonus Effects Model (BEM)
NPSTOE6	—	Proportion of YOS 0 cell that has a 6-year term of enlistment (TOE)
PTRIAL	J,YOS	Initial (trial) value for the fraction of those with grade J and YOS that will be promoted in the next year (variable is 0 for grade E-3)
PRATE3	YETS,YOS	Fraction of E-3 inventory in the YOS and YETS cell that will be promoted during the next year.
SCONTRT	J,YETS,YOS,C	Fraction of the inventory that will still be in the Air Force a year later out of those in the cell at the beginning of the year
SELECT	J	Fraction of airmen with grade J who will be promoted each year
SRELNGS	J,YETS,YOS,C	Fraction that will reenlist during a year out of those who remain in the Air Force at the end of the year and who began the year in the cell Note: SRENLGS(J,YETS,YOS,C) = 0 for all YETS > 1
STSAMA	—	Fraction of career force entrants from the AFSC who will remain in the same occupation if no bonus is offered
TOE6A	J,YETS,YOS,C	Probability that a person will choose a 6-year TOE given that he reenlists from cell J,YETS,YOS,C and does not receive a bonus
TOE6B	Z	Coefficient of the bonus incentive variable in the probability that a person will choose a 6-year TOE when some bonus is offered, given that he reenlists in zone Z
TOE6C	Z	Increase in the probability that a person will choose a 6-year TOE when some bonus is offered, given that he reenlists in zone Z

Cardinal of AF/DPXA who showed that promotion flows are well described by a simple model. A tentative or "trial" promotion rate is assigned to each combination of grade, year of service, and AFSC group.[1] The IPM promotion rate is then the multiple of this trial rate resulting in the user-specified total promotion count (or, equivalently, grade-specific end strength). The input array, PTRIAL, gives the trial promotion rates for the basic AFSC underlying the self-sustaining ladder. The steady-state algorithm is slightly more complicated but results in the same distribution of promotions across grade and YOS as in the trial rates and in the IPM.

Loss and Reenlistment Rates

The middle-term loss model provides estimates of the fraction of persons in each cell who will leave the Air Force in any given year and the fraction of those who choose to stay who will do so by extending rather than by reenlisting. The continuation rate, SCONTRT, stored by the model is equal to 1 minus the loss rate. The reenlistment rate, SRENLGS, is 1 minus the extend-given-stay rate. The subscripts for SCONTRT and SRENLGS are those that define cells in one year's inventory for one AFSC in the Middle-Term Disaggregate IPM. Array SRENLGS is set to 0 for YETS ≥ 2.

The loss and extension rates used for the model should be calculated assuming no bonus is offered and should not be based on extreme economic conditions. They should be blended (transformed from a cohort rate to a fiscal year rate)[2] using the assumption that reenlistments cannot occur before ETS. The same loss and extension rates are used for the BEM and for the YOSTG.

Within the initialization routine, the continuation and reenlistment rates that can be affected by bonuses are stored in additional arrays, ALPHA and ALPHAR, with the same dimensions as SCONTRT. These arrays are used along with the arrays BETA and BETAE to calculate the values of SCONTRT and SRENLGS that will result from any specified value of the decision variables. The equations used for these calculations are described in the subsection below entitled "Adjusting Flow Rates."

CAREERS Program Flow

The CAREERS program flows are modeled as a two-part decision. The first part gives the probability that a person entering the career force will remain in the same AFSC. This

[1] Eakle-Cardinal used year-in-grade as an additional subscript. Although this is certainly a powerful predictor of promotion rates, it was decided that the costs involved in maintaining an extra dimension in the DMI outweighed the benefits.

[2] Cohort rates are tied to events in an airman's career rather than to calendar time.

probability is a linear function of the bonus offer in the AFSC with constant term STSAMA and slope STSAMB. The second decision is made only by persons who have decided to change AFSCs and is the choice of which AFSC to enter.

The number of persons who will enter the particular specialty being analyzed depends on the bonus offers in all occupations, but it may be approximated as a function of the bonus offer in this specialty. The Bonus Effects Model (BEM) calculates this approximation, which is entered into the array CAREERS. The details of the calculation and an assessment of the accuracy of the approximation may be found in Carter et al., 1988.

Terms of Enlistments

The proportion of first termers who are in a 6-year term of enlistment (TOE) is entered as a policy variable, NPSTOE6. All other first termers in the model are in 4-year TOEs.

The probability that a reenlistee chooses a 6-year term rather than a 4-year term is given by the models developed by Carter and Hackett[3] and implemented in the DMI. These probabilities depend on career field group, YOS, grade, YETS, and the bonus offer. Within each cell of the model, the probability depends only on the bonus offer. The linear form of that model (rather than the logistic) is used here, with constant terms given by TOE6A, the slope on the bonus incentive given by TOE6B, and the effect of some positive bonus given by TOE6C.

ADJUSTING FLOW RATES

A subroutine, called AFLOW in this documentation, adjusts flow rates based on the values of the input a; ay SSB(Z). The flow rates to be adjusted consist of the ETS continuation rates, reenlistment rates, choice of TOE rates, and occupational choice under the CAREERS program.[4] When SSB(Z) is between 0 and 6, AFLOW estimates the flow rates that would result from a bonus of that multiple. When SSB(Z) is less than 0 (or greater than 6), the effect of the bonus on the continuation and reenlistment rates is extrapolated, and the choice of TOE rates and CAREERS flow are set at the bonus multiple 0 (bonus multiple 6) level.

The input variables to AFLOW consist of: SSB, ALPHA, ALPHAR, BETA, BETAE, BETALB, CAREERS, STSAMA, STSAMB, BONPAY, BONCAP, TOE6A, TOE6B, and TOE6C. The arrays SCONTRT and SRENLGS are passed to AFLOW and returned with

[3] The models are briefly described in Walker et al., 1991.

[4] The prototype uses a less accurate means of estimating CAREERS flow than is recommended here.

adjusted values. These variables are described in Tables 4 and 5. The remaining output variables from AFLOW are CAREST and PTOE6, described in Table 6.

An internal data array ZONE is dimensioned by YOS. ZONE(YOS) gives the bonus zone for a reenlistee in that YOS category.

Continuation and Reenlistment Rates

The continuation rates and reenlistment rates are adjusted using the coefficients found in the latest fitting of the middle term loss models. Note that the coefficients BETA, BETAE, and BETALB refer to the loss models; consequently their effect in calculating continuation and reenlistment rates is subtracted, not added. A linear specification is recommended.[5] The adjustments use coefficients from the cohort models to adjust the

Table 5

Variables Calculated During Initialization Procedure

Name	Subscripts	Meaning
ALPHA	J,YETS,YOS,C	Fraction of the inventory that will still be in the Air Force a year later out of those in the cell at the beginning of the year, assuming bonus = 0 (initialized to values of SCONTRT array)
ALPHAR	J,YETS,YOS,C	Fraction that will reenlist during a year out of those who remain in the Air Force at the end of the year and who began the year in the cell, assuming bonus = 0 (initialized to values of SRENLGS array)
PMULT	J	Current multiplier for trial promotion rates (initialized to 1)
TENDSTR	—	Trained end strength (initialized to the sum of SSA(J) over all grades J)

Table 6

Variables Calculated by Adjusting Flow Rate Routine

Name	Subscripts	Meaning
CAREST	—	Number of persons entering the AFSC through CAREERS program
PTOE6	J,YETS,YOS,C	Probability that a person in the cell will choose a 6-year term of enlistment

[5] The current prototype uses a logit specification, but this complication does not appear to add value to the model. It may even detract from the model's usefulness because it limits continuation rates and reenlistment rates to be less than 1; a user might wish to use larger continuation rates to simulate the effect of an inflow of retrained airmen.

blended rates. See Carter et al. (1988) for a justification of this decision. The adjustment must account for the negative effect of earlier bonuses on later reenlistments.

The rates for reenlistment out of the first term are also adjusted to account for the fraction of career force entries from this self-sustaining ladder who will choose to enter other SSLs. This is done by multiplying the reenlist-given-stay rate by the fraction of persons estimated to reenlist into this specialty.

The exact code required to adjust the continuation and reenlistment rates may change if there are changes in the form in which the bonus effect enters the loss model. The following specification assumes that (1) bonus effects appear only in the YETS = 1 loss and extension equations; (2) bonus effects are linear; and (3) BETALB(Z) is a coefficient on a dummy variable for Z = 2, and BETALB(Z) is 0 for Z = 1 and 3. These assumptions are similar to the current loss model specification.

Because the code must be adapted to changes in the loss models, the following specification should be viewed as illustrative rather than definitive:

```
DO YOS = 3 TO 14
Z = ZONE(YOS)
DO OVER C
DO OVER J
SCONTRT(J,1,YOS,C) = ALPHA(J,1,YOS,C) – BETA(Z) × SSB(Z)
IF Z = 2 AND SSB(1) > 0 THEN SSCONTRT(J,1,YOS,C) =
                         SSCONTRT(J,1,YOS,C) – BETALB(Z)
IF C = 1 THEN DO
     STAYSAME = STSAMA + STSAMB × MIN(6,MAX(0,SSB(1)))
     SRENLGS(J,1,YOS,C) = (ALPHAR(J,1,YOS,C) – BETAE(Z) × SSB(Z)) × STAYSAME
     SRENLGS(J,0,YOS,C) = ALPHAR(J,0,YOS,C) × STAYSAME
     SRENLGS(J, – 1,YOS,C) = ALPHAR(J, – 1,YOS,C) × STAYSAME
     END
ELSE SRENLGS(J,1,YOS,C) = ALPHAR(J,1,YOS,C) – BETAE(Z) × SSB(Z)
END END END
```

CAREERS Flow Into Ladder

There are two aspects to the CAREERS flow. First, some of the career force entries from this SSL will reenlist into other SSLs. These flows have already been included by adjusting the first term reenlist-given-stay rate. Second, persons may enter this SSL through the CAREERS program. The total number of such persons is estimated and stored in the variable CAREST.

The array CAREERS gives the number of persons who will enter the basic AFSC through the CAREERS program in response to each integer bonus multiple. Simple linear interpolation is used to estimate the value for nonintegers between 0 and 6:

```
IF SSB(Z) ≤ 0 THEN CAREST = CAREERS(0)
ELSE IF SSB(Z) ≥ 6 THEN CAREST = CAREERS(6)
ELSE DO
    BM = FLOOR(SSB(Z))
    CAREST = CAREERS(BM) + (CAREERS(BM + 1) – CAREERS(BM)) ×
    (SSB(Z) – BM)
    END
```

Term of Enlistment Choice

Separate term of enlistment equations were fitted for each zone Coefficients that do not appear in a particular model are input as 0. So, for example, TOE6B(3) is input as 0 since no effect of the bonus incentive is found in the zone C model. The effect of bonus multiples is truncated to be between 0 and 6.

The specification processes each zone sequentially. It stores the truncated bonus multiple in local temporary variable B. It then stores the expected bonus incentive in the zone in local temporary variable BONINC. The proportion of persons in each cell who will choose a 6-year TOE is estimated and placed in the output array PTOE6:

```
DO YOS = 3 TO 13
Z  = ZONE(YOS)
B = MIN(6,MAX(0,SSB(Z)))
BONINC = MIN(BONCAP,6 × BONPAY(Z) × B) – MIN(BONCAP,4 × BONPAY(Z) × B)
IF SSB(Z) > 0 THEN SOMBON = 1
        ELSE SOMBON = 0
DO OVER C
DO OVER J
DO YETS = -1 TO 1
PTOE6(J,YETS,YOS,C) = TOE6A(J,YETS,YOS,C) + TOE6B(Z) × BONINC +
                    TOE6C(Z) × SOMBON
```

END END END END

CALCULATING THE STEADY-STATE SOLUTION

Jacquette et al. (1977) discuss two solution methods for steady-state algorithms—those that start with accessions and project forward, and those that start with the highest grade and calculate backward to the number in earlier grades. In this problem promotion rates are known rather than number in grade, the forward calculation is the solution method chosen.

Figure 3 is a flow chart of the routine that calculates the steady-state inventory from the input values of the flow rates. The algorithm traces the fraction of persons who are expected to be found in each cell for each person who is present in the YOS 0 cell. The grades are processed sequentially, with an iterative procedure used to obtain the correct promotion rate. The rules for processing each grade are called the spreading equations. After the highest grade has been spread, the force is multiplied by constants that will produce the correct trained end strength and the correct balance of input from NPS accessions and from the CAREERS program.[6] This multiplication process is called normalizing the force.

The input to the steady-state calculations include all the flow rates that were adjusted for the effects of the decision variables (PTOE6, SCONTRT, SRENLGS, and CAREST), plus PTRIAL, TENDSTR, and STUDENTFR. The array PMULT is input and modified by the program so that successive calls will start with the values chosen by the previous call. Table 7 gives the major variables calculated in the program. The arrays SSF, PROM, RENL, and PMULT are returned to the calling program for output purposes.

The spreading equations implement the basic rules of steady state. A steady-state system is one where the rate of flow into each cell equals the rate of flow out of the cell. In this case, since one of the dimensions of each cell is YOS, the annual flow out of each cell must equal the inventory in that cell. Thus we can restate the condition for steady state as:

$$\text{inventory(cell)} = \text{annual flow into cell.}$$

The flow into a cell can be any of five types, depending on whether during the previous year the person had been:

1. Neither promoted nor reenlisted.
2. Promoted, but not reenlisted.

[6] It would be possible to allow input at a later career stage by extending the current algorithm, but this is not done here.

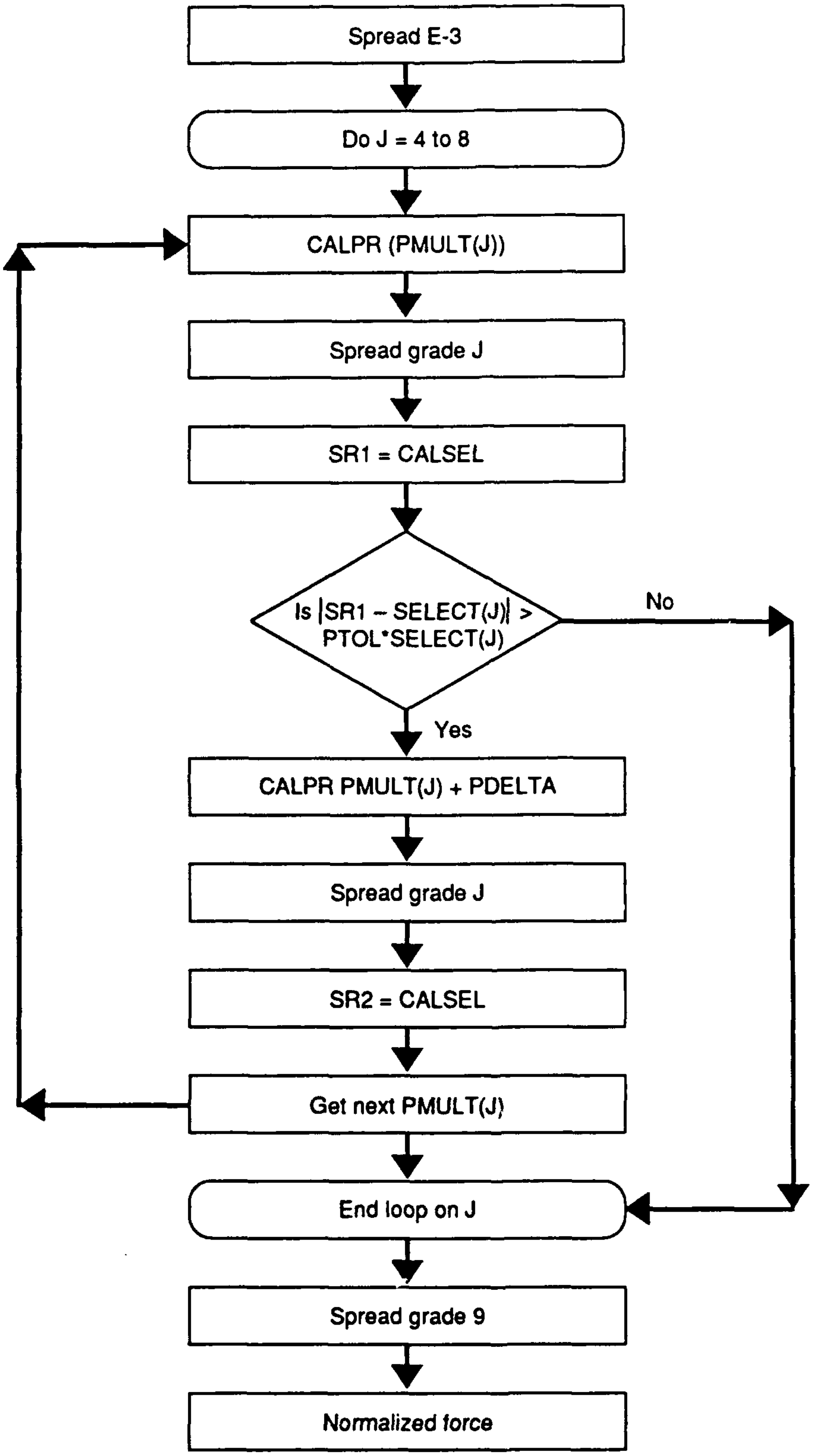

Fig. 3—Calculating the steady-state solution

Table 7

Major Variables in Steady-State Solution

Name	Subscripts	Meaning
SSF	J,YETS,YOS,C	Steady-state force in cell
PROM	J,YETS,YOS,C	Promotions out of grade J that will appear in the grade J+1,YETS,YOS,C cell
RENL	J,YETS,YOS,C	Reenlistments out of grade J that will appear in either the cell J+1,YETS,YOS,C or in the cell J,YETS,YOS,C. Note that RENL(3,YETS,YOS,C) is initialized to 0, and RENL(J,YETS,YOS,4) is never calculated
PTOL	—	Maximum tolerance allowed in the difference between proportion of grade promoted and desired proportion as a percent of desired proportion (initialized to a constant—the prototype uses .001)
PDELTA	—	Increase in multiplier for promotion rates used to obtain derivative of select rate with respect to multiplier (initialized to a constant—the prototype uses .05)
PRATE	J,YOS	Fraction of those with grade J and YOS who will be promoted in the next year
PMULT	J	Current multiplier for trial promotion rates

3. Reenlisted, but not promoted.
4. Both reenlisted and promoted.
5. Accessed without prior service.

The remaining way of entering the self-sustaining ladder is through the CAREERS program. The simplifying assumption made here is that entry through the CAREERS program will occur only at the time of reenlistment into the career force. Thus these flows will be counted within groups 3 and 4 above.

The next three subsections detail the flow into each cell by calculating the expected inventory generated by a NPS accession rate of one person per year.

Grade E-3 Spreading Equations

Only two cells have positive inventory and YOS 0, and both are in grade E-3. Since we wish to find the fraction of persons who are expected to be found in each cell out of a single person who is present in YOS 0 cell, the definition of NPSTOE6 means that:

$$\mathrm{SSF}(3,6,0,1) = \mathrm{NPSTOE6}, \text{ and}$$
$$\mathrm{SSF}(3,4,0,1) = 1 - \mathrm{NPSTOE6}.$$

To calculate SSF for other YETS in the same terms:

```
DO ITOE = 4 AND 6
DO YETS = ITOE – 1 TO – 1 BY – 1
YOS = ITOE – YETS
SSF(3,YETS,YOS,1) = SSF(3,YETS + 1,YOS – 1,1) × SCONTRT(3,YETS + 1,YOS – 1,1)
                    × (1 – SRENLGS(3,YETS + 1,YOS – 1,1)) × (1 – PRATE3(YETS +
                    1,YOS – 1))
PROM(3,YETS,YOS,1) = SSF(3,YETS + 1,YOS – 1,1) × SCONTRT(3,YETS + 1,YOS – 1,1)
                    × (1 – SRENLGS(3,YETS + 1,YOS – 1,1)) × PRATE3(YETS + 1,YOS – 1)
END END
```

Since E-3s are not allowed to reenlist in the model, RENL(3,YETS,YOS,C) is initialized to 0 for all cells. This concludes the spreading calculations for grade E-3.

Grades E-4 Through E-8 Spreading Equations

The spreading calculations are similar for grades E-4 through E-8, with the calculations for each preceding grade being completed before beginning the next grade.[7] The order of the calculations is important, since it is necessary to have completed the calculations of all variables used on the right-hand side of each equation before the equation is used. The description will cover the calculations for a particular grade J.

For category of enlistment 1, the only flows into the cell are promotions from the grade below and continuations of persons in the same grade who are neither promoted nor reenlisted:

```
DO ITOE = 4 AND 6;
DO YETS = ITOE – 1 TO – 1 BY – 1;
YOS = ITOE – YETS;
SSF(J,YETS,YOS,1) = PROM(J – 1,YETS,YOS,1)
                    + SSF(J,YETS + 1,YOS – 1,1) × SCONTRT(J,YETS + 1,YOS – 1,1)
                     × (1 – SRENLGS(J,YETS + 1,YOS – 1,1)) × (1 – PRATE(J,YOS – 1))
PROM(J,YETS,YOS,1) = SSF(J,YETS + 1,YOS – 1,1) × SCONTRT(J,YETS + 1,YOS – 1,1) ×
                    × (1 – SRENLGS(J,YETS + 1,YOS – 1,1)) × PRATE(J,YOS – 1)
END END
```

[7] It is possible to eliminate some calculations by limiting the ranges through which the calculations are performed. For example, the first term does not contain persons in the highest grades. We simplify the discussion by ignoring these savings, although some are actually achieved in the current prototype.

For categories of enlistment 2 and 3, flows into the cells with YETS = 4 and YETS = 6 include reenlistments as well as promotions and continuations of those who are neither promoted nor continued. The reenlistees can of course be promoted in the same year as the reenlistment. Just as for the first term, the calculations follow a cohort as it ages through to the end of each possible term.

In general, we can calculate:

$$\begin{aligned}\mathrm{RENL}(J,6,\mathrm{YOS},C) = \Sigma\ \{&\mathrm{SSF}(J,m,\mathrm{YOS}-1,c) \times \mathrm{SCONTRT}(J,m,\mathrm{YOS}-1,c)\\ &\times \mathrm{SRENLGS}(J,m,\mathrm{YOS}-1,c) \times \mathrm{PTOE6}(J,m,\mathrm{YOS}-1,c)\}\end{aligned} \tag{3.2}$$

where the range of the summation depends on C. For C = 2, the sum is over m = – 1,0, and 1 and c = 1. For C = 3, the sum is over m = – 1,0, and 1 and c = both 2 and 3. Similarly,

$$\begin{aligned}\mathrm{RENL}(J,4,\mathrm{YOS},C) = \Sigma\{&\mathrm{SSF}(J,m,\mathrm{YOS}-1,c) \times \mathrm{SCONTRT}(J,m,\mathrm{YOS}-1,c)\\ &\times \mathrm{SRENLGS}(J,m,\mathrm{YOS}-1,c) \times [1-\mathrm{PTOE6}(J,m,\mathrm{YOS}-1,c)]\}\end{aligned} \tag{3.3}$$

where the definition of the range is identical to that for the case of YETS = 6.

The order of the spreading calculations follows. The variable Y ranges over all years of service that can be the first year of a term in category of enlistment C.

```
DO C = 2 AND 3.
IF C = 2 THEN DO:
        LOW =  2
        HI =  8
        END
ELSE DO
        LOW =   6
        HI = 19
        END
DO Y = LOW TO HI
YOS =Y
Calculate RENL(J,6,YOS,C) from formula 3.2
$SSF(J,6,YOS,C) = PROM(J-1,6,YOS,C) + RENL(J,6,YOS,C) \times (1-PRATE(J,YOS-1))$
$PROM(J,6,YOS,C) = RENL(J,6,YOS,C) \times PRATE(J,YOS-1)$

YOS = Y + 1
```

```
IF YOS ≤ 19 THEN SSF(J,5,YOS,C) = PROM(J – 1,5,YOS,C)
        + SSF(J,6,YOS – 1,C) × SCONTRT(J,6,YOS-1,C) × (1 – PRATE(J,YOS – 1))
IF YOS ≤ 19 THEN PROM(J,5,YOS,C) =
       SSF(J,6,YOS – 1,C) × SCONTRT(J,6,YOS – 1,C) × PRATE(J,YOS – 1)

YOS = Y + 2
Calculate RENL(J,4,YOS,C) from formula 3.3
IF YOS ≤ 19 THEN SSF(J,4,YOS,C) = PROM(J – 1,4,YOS,C)
                        + RENL(J,4,YOS,C) × (1 – PRATE(J,YOS – 1))
         + SSF(J,5,YOS – 1,C) × SCONTRT(J,5,YOS – 1,C) × (1 – PRATE(J,YOS – 1))
IF YOS ≤ 19 THEN PROM(J,4,YOS,C) =
        (SSF(J,5,YOS – 1,C) × SCONTRT(J,5,YOS – 1,C)
        + RENL(J,4,YOS,C)) × PRATE(J,YOS – 1)

DO YETS = 3 TO – 1 BY – 1
YOS = Y + 6 – YETS
IF YOS ≤ 19 THEN SSF(J,YETS,YOS,C) = PROM(J – 1,YETS,YOS,C)
            + SSF(J,YETS + 1,YOS – 1,C) × SCONTRT(J,YETS + 1,YOS – 1,C)
             × (1 – SRENLGS(J,YETS + 1,YOS – 1,C)) × (1 – PRATE(J,YOS – 1))
IF YOS ≤ 19 THEN PROM(J,YETS,YOS,C) = SSF(J,YETS + 1,YOS – 1,C)
            × SCONTRT(J,YETS + 1,YOS – 1,C) ×
            × (1 – SRENLGS(J,YETS + 1,YOS – 1,C)) × PRATE(J,YOS – 1)
END END END
```

Category of enlistment 4 contains all persons with YOS greater than 19. The loss rates for years of service 19 and greater do not depend on YETS. For convenience, all retirement eligible persons are stored in cells with YETS = 1 and the continuation rate for persons with YOS greater than or equal to 19 is accessed from the cell with YETS = 1. The temporary variable R(J) is the total of SSF summed over all cells with YOS = 19 and grade J. Then:

```
SSF(J,1,20,4) = R(J – 1) × SCONTRT(J – 1,1,19,4) × PRATE(J – 1,19)
         + R(J) × SCONTRT(J,1,19,4) × (1 – PRATE(J,19))
PROM(J,1,20,4) = R(J) × SCONTRT(J,1,19,4) × PRATE(J,19)
```

```
DO FOR YOS = 21 TO 29
SSF(J,1,YOS,4) = SSF(J – 1,1,YOS – 1,4) × SCONTRT(J – 1,1,YOS – 1,4) ×
PRATE(J – 1,YOS – 1) + SSF(J,1,YOS – 1,4) × SCONTRT(J,1,YOS – 1,4) ×
(1 – PRATE(J,YOS – 1))
PROM(J,1,YOS,4) = SSF(J,1,YOS – 1,4) × SCONTRT(J,1,YOS – 1,4) × PRATE(J,YOS – 1)
END
```

This concludes the description of the spreading calculations for the grades E-4 through E-8. Having completed a particular grade J, control passes to the section of code that evaluates whether the promotions achieved are equal to the select rate. Those calculations will be given after the calculations for grade E-9.

Grade E-9 Spreading Equations

The calculations for E-9 are simpler than those for earlier grades, because there are no promotions out of E-9, and there are no first term calculations. The calculations follow where J = 9:

```
DO C = 3
DO Y = 0 to 19
YOS = Y
Calculate RENL(J,6,YOS,C) from formula 3.2.
SSF(J,6,YOS,C) = PROM(J – 1,6,YOS,C) + RENL(J,6,YOS,C)
YOS = Y + 1
IF YOS ≤ 19 THEN SSF(J,5,YOS,C) = PROM(J – 1,5,YOS,C)
                        + SSF(J,6,YOS – 1,C) × SCONTRT(J,6,YOS – 1,C)
YOS = Y + 2
Calculate RENL(J,4,YOS,C) from formula 3.3
IF YOS ≤ 19 THEN SSF(J,4,YOS,C) = PROM(J – 1,4,YOS,C)
                + RENL(J,4,YOS,C) + SSF(J,5,YOS – 1,C) × SCONTRT (J,5,YOS – 1,C)
DO YETS = 3 TO  – 1 BY – 1
YOS = Y + 6 – YETS
IF YOS ≤ 19 THEN SSF(J,YETS,YOS,C) = PROM(J – 1,YETS,YOS,C)
                  + SSF(J,YETS + 1,YOS – 1,C) × SCONTRT(J,YETS + 1,YOS – 1,C)
    × (1 – SRENLGS(J,YETS + 1,YOS – 1,C))
    END END END
```

Calculate R(J) = the total inventory in grade 9 and YOS 19. Then:

$$SSF(J,1,20,4) = R(J-1) \times SCONTRT(J-1,1,19,4) \times PRATE(J-1,19) + R(J) \times SCONTRT(J,1,19,4)$$

```
DO YOS = 21 TO 29
SSF(J,1,YOS,4) = SSF(J – 1,1,YOS – 1,4) × SCONTRT(J – 1,1,YOS – 1,4) ×
PRATE(J – 1,YOS – 1) + SSF(J,1,YOS – 1,4) × SCONTRT(J,1,YOS – 1,4)
END
```

Determining Promotion Rate Parameters

This subsection describes the subroutines for determining PRATE(J,YOS), the promotion rates for each year of service that will produce an overall promotion rate equal to SELECT(J). The subroutines process data for only one grade at a time. Three subroutines are involved: (1) CALPR calculates the promotion rates for a given value of the multiplier on the input trial promotion rates, (2) CALSEL calculates the select rate observed from the results of a previous call to the routine that evaluated the spreading equations for grade J using the current PRATE value, and (3) GETNP estimates the value of the multiplier that will produce a promotion rate equal to SELECT(J). The order of the calls to these routines is shown in Fig. 3.

The input to CALPR consists of J,PTRIAL, and the multiplier XMULT. As shown in Fig. 3, XMULT is equal to either PMULT(J) or PMULT(J) + PDELTA, depending on where CALPR is called. The subroutine fills in the array PRATE for the input grade J and for all YOS using the formula:

$$PRATE(J,YOS) = XMULT \times PTRIAL(J,YOS).$$

CALSEL is a function. The input to CALSEL consists of J, the results of the spreading equations for grade J in array SSF, and the promotion out of grade J in array PROM. CALSEL calculates the temporary variables:

$$X = \Sigma\,(SSF(J,m,y,c)),$$
$$Y = \Sigma\,(PROM(J,m,y,c)), \text{ and}$$
$$Z = \Sigma\,PROM\,(J\text{-}1,m,y,c)$$

where the sums are over all YETS, YOS, and categories of enlistment. The observed select rate, SR, is returned where:

SR = Y/(X – Z).

The subroutine GETNP estimates the value of the multiplier that will produce a promotion rate equal to SELECT(J). The input consists of SELECT(J), PMULT(J), PDELTA, and the select rates that result from using a multiplier of PMULT(J) and a multiplier of PMULT(J) + PDELTA stored in SR1 and SR2 respectively. The routine returns a new value of PMULT(J).

GETNP uses the Newton-Rhapson iteration technique, which in this application converges very quickly. The derivative of the select rate with respect to the multiplier is given by:

DERV = (SR2 – SR1)/PDELTA.

And the next value of the multiplier is then given by:

PMULT(J) = PMULT(J) + (SELECT(J) – SR1)/DERV.

Normalizing the Force

The process of normalizing the force consists of multiplying the cells in the array SSF so that the total of SSF equals the desired total end strength, and the number of career force entrants equals the sum of the number expected to enter the SSL through the CAREERS program and from the first term force.

We begin the normalizing calculations after the process of spreading the force is complete. At this time, the array SSF contains numbers proportional to the number of persons who would be in each cell if the only input to the SSL were NPS accessions. The portion of the array SSF that corresponds to categories of enlistment 2, 3, and 4 give the number of persons who would be in each cell in the career force if, as is assumed by the model, the only input to the career force of the SSL came at entry into the career Air Force.

To understand the normalizing calculations, consider the following totals:

$$ST = \Sigma\, SSF(j,m,y,c), \tag{3.4}$$

$$SCT = \Sigma\, SSF(j,m,y,c), \text{ and} \tag{3.5}$$

$$SCE = \Sigma\ RENL(j,m,y,2), \tag{3.6}$$

where all sums are over all grades, YETS, and YOS, and the first sum is over all categories of enlistment, but the middle sum is over only the career force defined as c = 2, 3, and 4. For each increase of one in the expected number of annual career force entrants, the career force would increase by SCT/SCE persons. At any one time, the number of the graduates of the CAREERS program that are in the self-sustaining ladder is given by:

$$CARCOUNT = CAREST \times SCT/SCE.$$

All other members of the ladder have entered it as NPS accessions. Of those currently in the YOS = 0 cell, the fraction STUDENTFR are not yet trained and therefore cannot fill a trained personnel authorization. For each increase of one in the size of the YOS = 0 cell, the trained inventory will increase by ST – STUDENTFR persons. The total trained inventory should equal TENDSTR. This will occur if each cell in the first term force is multiplied by:

$$(TENDSTR - CARCOUNT)/(ST - STUDENTFR),$$

and each cell in the career force by:

$$(TENDSTR - CARCOUNT)/(ST - STUDENTFR) + CAREST/SCE.$$

In order to see this, note that after these multiplications, the total inventory will equal:

$$(TENDSTR - CARCOUNT) \times ST/(ST - STUDENTFR) + CARCOUNT,$$

and the untrained inventory will equal:

$$(TENDSTR - CARCOUNT)/(ST - STUDENTFR) \times STUDENTFR.$$

Subtracting the untrained inventory from the total inventory yields the trained inventory of size TENDSTR.

For the convenience of the reader, I repeat these results in the form of an algorithm to normalize the force. In addition to normalizing the inventory array SSF, the algorithm also normalizes the number of reenlistments and promotions for subsequent printing and output.

The algorithm deals with the unusual case (which should never arise) where so many CAREERS graduates are estimated to exist that the SSL would be overstrength with less than one annual NPS accession. The algorithm follows.

Calculate ST, SCT, and SCE from Eqs. 3.4 through 3.6.
Calculate:

```
CARCOUNT = Minimum(CAREST × SCT/SCE, TENDSTR – ST),
CAREST   = SCE × CARCOUNT/SCT,
X1 = (TENDSTR – CARCOUNT)/(ST – STUDENTFR), and
X2 = (TENDSTR – CARCOUNT)/(ST – STUDENTFR) + CAREST/SCE.
DO J = 3 to 9
  DO YETS = – 1 to 6
    DO YOS = 0 to 30
      SSF(J,YETS,YOS,1) = X1  × SSF(J,YETS,YOS,1)
      PROM(J,YETS,YOS,1) = X1  × PROM(J,YETS,YOS,1)
      DO C = 2,3, AND 4
        SSF(J,YETS,YOS,C) = X2  ×  SSF(J,YETS,YOS,C)
        PROM(J,YETS,YOS,C) = X2  ×  PROM(J,YETS,YOS,C)
        IF C NE 4 THEN RENL(J,YETS,YOS,C) = X2  ×  RENL(J,YETS,YOS,C)
END END END END
```

CHOOSING IMPROVED VALUES OF BONUS VARIABLES

This subsection describes the algorithm that controls the optimizing calculations. The methodology is steepest descent. As shown in Fig. 4, the algorithm can be implemented as a series of subroutines, the first of which calculates the current value of the penalty function, which it stores in the variable COST. The second subroutine calculates the gradient, the vector giving the derivative of the penalty function with respect to the three decision variables. Movement in the direction opposite to the gradient will result in the largest decrease in the penalty function.

The third subroutine determines how far we can go in the minimizing direction without running into a constraint. Let X be this farthest point. The rest of the optimizing algorithm selects the point on the straight line between the current position and X where it estimates that the minimum value of the penalty function will occur. Define the scalar variable LAMDA as the distance between the current position and any point on this straight

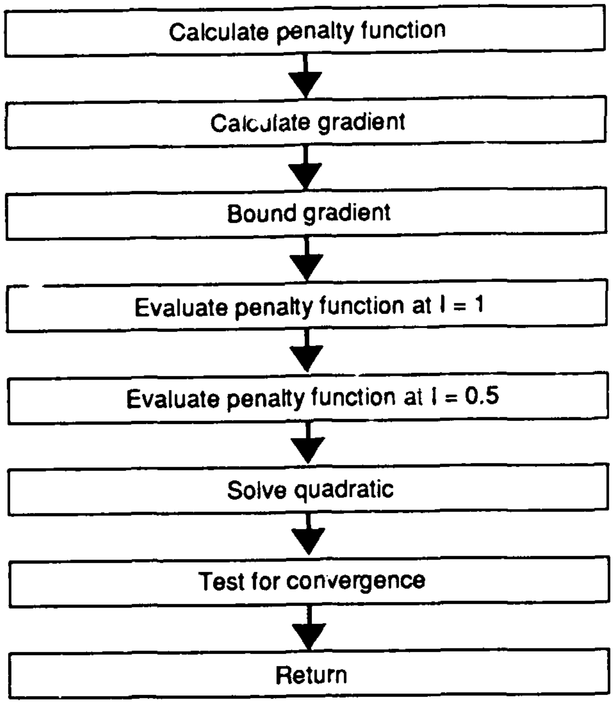
Calculate penalty function
Calculate gradient
Bound gradient
Evaluate penalty function at l = 1
Evaluate penalty function at l = 0.5
Solve quadratic
Test for convergence
Return

Fig. 4—Obtaining improved values of SSB

line using a scale so that the value of LAMDA at X is 1. The penalty function is evaluated at X and at the point where LAMDA = 0.5 (halfway between the current position and X). Including the current position, the penalty function is then known at three points on a straight line. A quadratic is fitted in LAMDA through these three points and the solution is that point on the line section where the quadratic is minimized.

Below is a description of the subroutine that calculates the penalty function. The subroutine EVAL, which is called twice in Fig. 4, is also called within the gradient subroutine. It is described next. Then I will give specifications for the gradient calculations, the bounding calculations, the routine that obtains the minimum value of the quadratic equation, and the test for convergence.

Subroutine EVAL

This routine accepts as input a particular set of values for the decision variables, SSB, and returns the value of the penalty function in the argument SSOBJ. To do this, it first calls the routine AFLOW, which adjusts the flow rates to correspond to the input values of the decision variables. All the variables known to AFLOW must either be passed into this subroutine or defined in a common area. Then it calls the routine that finds the steady-state inventory corresponding to the adjusted flow rates. In this call locally defined arrays are used for the variables calculated in the steady-state program (SSF, PROM, and RENL) to maintain the current steady-state solution for other purposes.

Finally EVAL calls the routine that actually calculates the penalty function, CALPEN, and passes its result back to the calling routine.

Subroutine CALPEN

This routine calculates the penalty function. The input consists of a steady-state inventory in array SSF, the authorization array SSA(J), and the weight array W(J). The output is the objective function of Eq. (3.1), which is stored in the argument SSOBJ. The subroutine begins by calculating the number of inventory in each grade:

$$\mathrm{SSFG}(J) = \Sigma\,(\mathrm{SSF}(J,m,y,c)),$$

where the sum is over all YETS, YOS, and categories of enlistment. The calculation of the objective function is then simply the application of Eq. (3.1).

Subroutine GRADIENT

The gradient is computed numerically. The input to the gradient calculation consists of the current value of the penalty function in variable COST; the array SSB, which contains the current values of the decision variables; and the flow rate data required to be passed to EVAL. The subroutine must also be able to pass SSA and W to CALPEN. The output of the routine is the gradient stored in DB(Z), for Z = 1, 2, 3.

The subroutine initializes an array, DUPSSB(Z), to be equal to SSB. This array is then passed to EVAL in place of SSB in subsequent calls:

```
DO Z = 1 TO 3
DUPSSB(Z) = SSB(Z)
END
DO Z = 1 TO 3
DUPSSB(Z) = SSB(Z) + .05
CALL EVAL
DB(Z) = (SSOBJ – COST)/.05
DUPSSB(Z) = SSB(Z)
END
```

Subroutine BOUND

This routine finds the maximum amount that one can move in the direction opposite the gradient before encountering one of the constraints. The result is returned in the scalar variable R. The input variables consist of (1) SSB, the current values of the decision variables; (2) DB, the gradient vector; (3) COST, the current value of the penalty function, (4) BBOT, the lower bounds on the decision variables; and (5) BTOP, the upper bounds on the decision variables.

The algorithm begins by considering whether any of the decision variables is already at its limit. If so, the derivative in that direction is set to zero so that improvement in the remaining decision variables may be sought. Then it tentatively establishes the limit at the minimum point along the gradient. Then it examines each constraint in turn and tightens the limit as needed.

The algorithm follows:

```
DO Z = 1 TO 3
    IF DB(Z) < 0 AND SSB(Z) = BTOP(Z) THEN DB(Z) = 0;
    IF DB(Z) > 0 AND SSB(Z) = BBOT(Z) THEN DB(Z) = 0;
END

SUMG = 0;
DO Z = 1 TO 3
    SUMG = SUMG + DB(Z) × DB(Z)
END
R = COST/SUMG
DO Z = 1 TO 3
    IF DB(Z) < 0 AND R > – (BTOP(Z) – SSB(Z))/DB(Z)
                    THEN R = – (BTOP(Z) – SSB(Z))/DB(Z)
    IF DB(Z) > 0 AND R > (SSB(Z) – BBOT(Z))/DB(Z)
                    THEN R = (SSB(Z) – BBOT(Z))/DB(Z)
END
```

Determining the Optimal Value

The next steps are to find the value of the penalty function at the extreme point and at the halfway point. Using the following steps, these results are stored in X1 and XH respectively. The array DUPSSB is again used to pass values of the decision variable to EVAL:

```
DO Z = 1 TO 3
DUPSSB(Z) = SSB(Z) – R × DB(Z)
END
CALL EVAL
X1 = SSOBJ
DO Z = 1 TO 3
DUPSSB(Z) = SSB(Z) – 0.5 × R × DB(Z)
END
CALL EVAL
XH = SSOBJ
```

At this point the value of the penalty function at three points is in the variables COST, XH, X1, corresponding to three values of LAMDA, 0, 0.5, and 1.0 respectively. The first three statements below obtain the value of LAMDA at the minimum point of a quadratic curve fitted through these three points. The *next two statements* check to make sure that the result remains within the constraints. The last line assumes that the extreme point is a minimum rather than a maximum. (If a maximum, the model assumes convergence has been reached.) The remaining steps store the next values of the decision variables in array NSSB.

```
LAMDA1 = – (4 × XH – X1 – 3 × COST)
LAMDA2 = (2.0 × (2.0 × X1 + 2.0 × COST – 4 × XH))
LAMDA = LAMDA1/LAMDA2
IF LAMDA > 1 THEN LAMDA = 1
IF LAMDA < 0 THEN LAMDA = 0
IF LAMDA2 < 0 THEN LAMDA = 0
DO Z = 1 TO 3
NSSB(Z) = SSB(Z) – LAMDA × R × DB(Z)
END
```

Convergence Test

The remaining activity is to determine whether the suggested changes in the decision variables are small enough to indicate convergence. The check is whether the difference in any variable exceeds the input variable STEPTOL

```
ICONV = 1
DO Z = 1 TO 3
IF ABS(SSB(Z) – NSSB(Z)) > STEPTOL THEN ICONV = 0
END
```

If ICONV remains = 1 at the end of the loop, or if the number of loops exceeds the variable MAXLOOP, the calculations for this self-sustaining ladder are complete and the solution is written out. Otherwise, the array SSB is set equal to NSSB and another iteration occurs, beginning with the call to AFLOW.

OUTPUT

The output from the steady-state optimizer has three purposes: (1) It provides information needed to create annual targets in the dynamic optimizer, (2) it is input into the AFSC target allocation module so that the user can examine the steady-state AFSC targets, and (3) it describes the steady-state target for the SSL, which may be of interest in itself. The first two purposes are served by the creation of computer files described in the first two subsections below. The third subsection describes additional statistics available from the model that might be either printed directly or fed to the target analysis module.

All output occurs following convergence. At this point the array SSB contains the optimal value of the decision variables; all flow rate variables and all the variables in the steady-state solution (Table 7) must be set to correspond to the optimal value of SSB. (These could be saved, or the program could insert additional calls to the subroutines that calculate the flow rates and the steady-state solution immediately before the output step.)

Creating Input for the Dynamic Optimizer

Table 8 lists the variables that are passed from the steady-state to the dynamic optimizer. The variables SSB and CAREST have previously been calculated. The value of SSOBJ corresponding to SSB was saved in the variable COST and can be retrieved from there for output.

As a simplification, the distribution of CAREERS retrainees is assumed in the model to correspond to the distribution of other career force entrants. We begin by calculating the

Table 8

Variables Sent from the Steady-State Optimizer to the Dynamic Optimizer

Name	Subscripts	Meaning
SSB	Z	Decision variables in steady-state solution
SSOBJ	—	Value of steady-state penalty function found at steady-state solution
CARDIST	J,YETS,YOS	Fraction of career force entries that join IPM cell (the category of enlistment is 2 by definition) Used to distribute CAREERS retrainees among IPM cells
CAREST	—	Inflow of CAREERS retrainees found in the steady-state solution
SSNPS	—	Number of persons in YOS = 0 cells in steady-state solution
SSCONT	Z	Steady-state number of persons continuing past ETS in the zone
SSENDSTR		Total end strength in steady-state including inventory in training

distribution of reenlistments into each IPM cell. For all grades $J > 3$, $YETS = 4$ and 6, all values of YOS, and categories of enlistment $C = 2$ and 3:

$$\begin{aligned}\mathrm{RENLIN}(J,\mathrm{YETS},\mathrm{YOS},C) = {} & \mathrm{RENL}(J,\mathrm{YETS},\mathrm{YOS},C) \times (1 - \mathrm{PRATE}(J,\mathrm{YOS}-1) + \\ & \mathrm{RENL}(J-1,\mathrm{YETS},\mathrm{YOS},C) \times \mathrm{PRATE}(J-1,\mathrm{YOS}-1).\end{aligned}$$

RENLIN is set to 0 for all other cells. Calculate the variable, TOTR, the number of reenlistments into category of enlistment 2:

$$\mathrm{TOTR} = \Sigma\ \mathrm{RENLIN}(J,\mathrm{YETS},\mathrm{YOS},2)$$

where the sum is over all grades, $YETS = 4$ and 6, and all YOS. Then:

$$\mathrm{CARDIST}(J,\mathrm{YETS},\mathrm{YOS}) = \mathrm{RENLIN}(J,\mathrm{YETS},\mathrm{YOS})/\mathrm{TOTR}.$$

The number of NPS accessions present at the end of each fiscal year is given by:

$$\mathrm{SNPS} = \mathrm{SSF}(3,4,0,1) + \mathrm{SSF}(3,6,0,1).$$

The number of persons continuing past ETS in each zone is given by:

$$SSCONT(Z) = \Sigma\ (SSF(J,1,YOS,C) \times SCONTRT(J,1,YOS,C)),$$

where the sum is over all grades J, categories of enlistment C, and the values of YOS in the zone.

The last value passed to the dynamic optimizer is SSENDSTR, which is the sum over all values of SSF.

The form of the output file can be determined to suit the ease of reading and writing the file. The only constraint is that separate records should be written for each self-sustaining ladder so that the model can be run on parts of the force.

Saving the Steady-State Solution

The entire force array SSF should be saved to a file in the form that is read by the AFSC target allocation module. This module also reads annual targets from the dynamic optimizer that require a year subscript. To enhance compatibility, it will be useful to add a distant year to the steady-state output file (say 30 years from the first year of the planning period).

Additional Output

The bonus effects model currently displays the number of reenlistments in each zone in the steady-state solution. This is found by summing the values of RENLIN over the appropriate YOS groups. Note that the YOS subscript in RENLIN gives the YOS at the end of the fiscal year containing the reenlistment. Thus for zone A, one sums over YOS = 4 to 6, rather than 3 to 5, and similar adjustments are made to the YOS in the other zones. The array RENLIN includes entrants to the self-sustaining ladder from the CAREERS program. Thus to obtain reenlistments out of the cohorts reaching ETS in zone A of the self-sustaining ladder, one can subtract the variable CAREST from the variable TOTR.

The exact form of additional output, if any, should be decided in consultation with users. However, one of the inputs to the target analysis module should be the entire array SSF. The input format used by the target analysis module should be the same as the input format used in the AFSC target allocation module so that the same output file can be input to both modules.

The user might also wish to see many of the variables passed to the dynamic optimizer either on a paper printout or displayed within the target analysis module. Of particular interest are the values of the decision variables (SSB), the number of NPS accessions (SSNPS), the number of entrants in the careers program (CAREST), and the total inventory

(SSENDSTR). The number of inventory in training may be calculated by subtracting trained end strength from SSENDSTR. Another variable that might be wanted is the fraction of persons who reach ETS in the zone who ever reenlist. This is merely the number of reenlistments in the zone divided by the number of decisionmakers who reach ETS in the zone, where the number of decisionmakers is given by:

$$DM(Z) = \Sigma\ (SSF(J,1,YOS,C)).$$

The sum is over all values of J and C and over the values of YOS in the zone.

IV. DYNAMIC OPTIMIZER

This section contains the mathematical specifications for the dynamic portion of the YOS target generator. Immediately below is an overview of the algorithm, including the order in which calculations occur. Subsequent subsections define the input data, how initial values are chosen for the decision variables, how flow rates are adjusted to reflect any given values of the decision variables, how the inventory is projected from these flow rates, how improved values of the decision variables are calculated from an approximation to the derivatives of the penalty function, and the criteria for convergence.

OVERVIEW

The dynamic portion of the YOSTG iteratively projects the inventory and chooses improved values for the decision variables. The decision variables are directly analogous to those of the steady-state target generator and are stored in the array B(Z,T) for Z = 1, 2, and 3 and T = 1 to PLANYEARS. They act as selective reenlistment bonus multiples when they are within the range from 0 to 6, and otherwise they just affect end of term loss and reenlistment decisions.

The inventory projection is over a horizon that is at most PLANYEARS + 30 years, but may be shorter if the inventory approaches close enough to its steady-state position. The period from T = 1 to PLANYEARS is called the planning horizon; the total duration of the inventory projection is called the projection period.

The optimal YOS target for each self-sustaining ladder is obtained separately. Figure 5 presents the flow for a single self-sustaining ladder. Each of the boxes represents a subroutine. The initialization subroutine reads input data that includes all the data read by the steady-state target generator, the initial inventory, and some information about the steady-state solution. The first inventory projection uses the steady-state values for each decision variable. In the next subroutine, this projection is used to obtain good values for the decision variables, called the "initial solution" in Fig. 5. Then the iterative loop is entered. In a call to the subroutine in the box labeled "Adjust flow rates," the loss rates, reenlistment rates, and term of enlistment proportions are adjusted to reflect the current values of the B array. The next step projects the inventory for as long as necessary until it approximates the steady-state solution.

In the following step, an improved value of the array B is sought. The convergence criteria are tested, and if the solution is deemed acceptable, the target inventory and flow

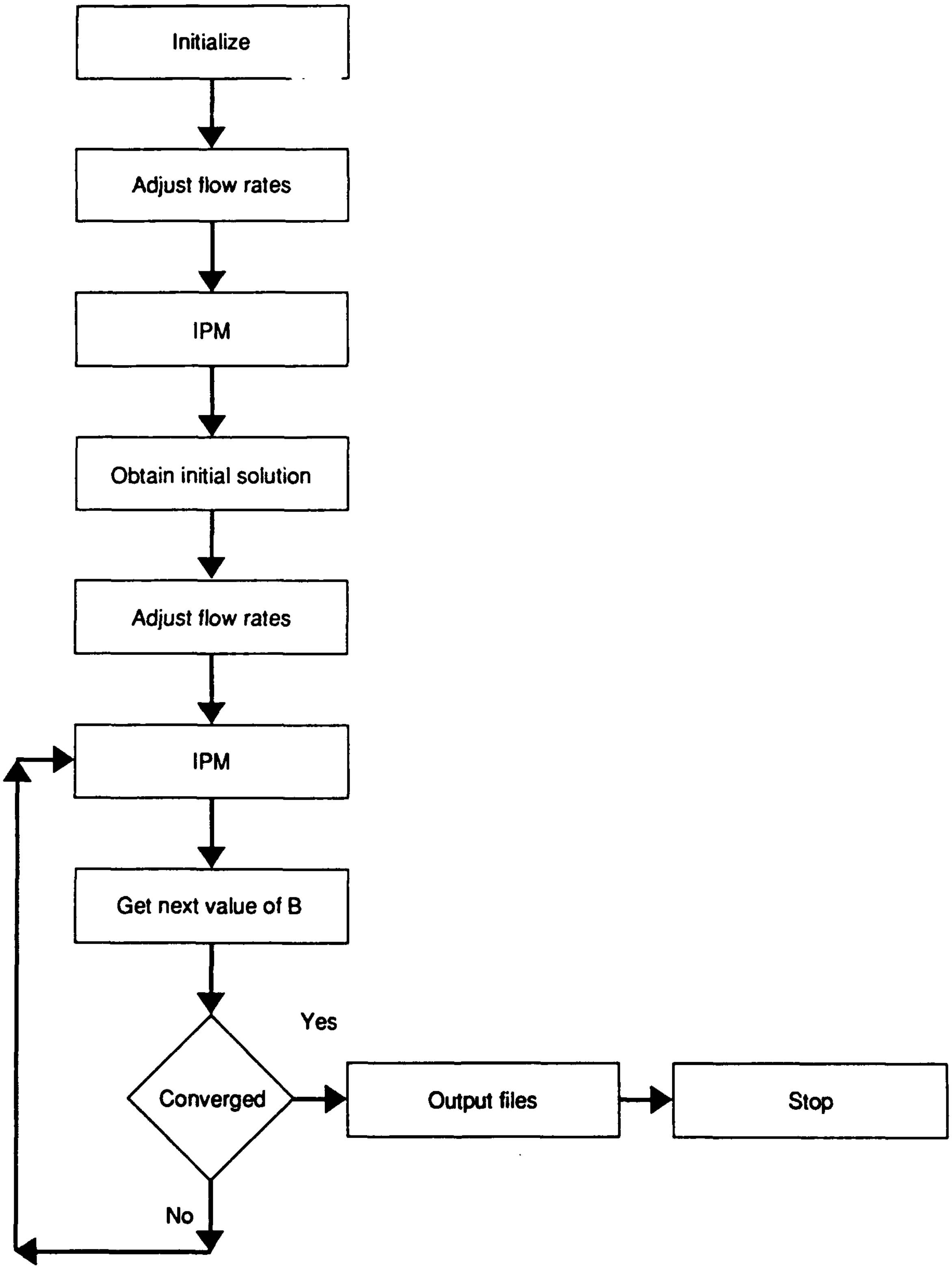

Fig. 5—Overview of dynamic model

data are output, and the run for this self-sustaining ladder ends. Otherwise, the iterative loop is repeated.

INITIALIZATION

The initialization subroutine of the module reads in the data needed for the run. This subsection discusses the subscripts used to index the data, the user specified input, and the data received from the steady-state target generator. Also discussed is the setting of initial values for some of the variables used in the computations.

Subscripts

Table 9 lists the subscripts that index data within the dynamic optimizer. Many of the subscripts are the same as those used in the steady-state optimizer. One addition, T, is used to keep track of the year of the inventory projection. The value of 0 is used for the initial inventory, so that the projection at the end of the first year corresponds to T = 1, etc. Another addition, CY (for cohort year), indexes just the planning horizon, which is at most nine years in the prototype. The other new subscript, H, is also a time variable and indexes the bonus history for the basic AFSC.

Control Variables

The first part of Table 10 lists the variables that enter the penalty function. If, for any given projection, the number of persons in grade J at the end of year T is FG(J,T), then the penalty assessed for year T is given by:

$$OBJT(T)=\sum_{J} W(J)\times[FG(J,T)-A(J,T)]^{2}. \tag{4.1}$$

Table 9

Subscripts for Dynamic Target Generator

Name	Range	Meaning
C	1 to 4	Category of enlistment
J	3 to 9	Grade
YETS	-1 to 6	Years before ETS
YOS	0 to 29	Years of service
Z	1 to 3	Bonus zone
BM	0 to 6	Bonus multiple
T	0 to 41	Projection year
H	-5 to 0	Years preceding T = 1
CY	0 to 11	Cohort year

Table 10

Input Variables for the Dynamic Optimizer

Name	Subscripts	Meaning
		Penalty Function
A	J,T	Authorizations by grade (from AUTHC of Sec. III. For T ≥ PLANYEARS, use the same authorizations as were used for the steady-state run)
W	J	Weight for grade in the penalty function
DISCOUNT	—	Discount factor
		Optimization Control Variables
STEPTOL	—	Tolerance factor in the convergence test
MAXLOOP	—	Maximum number of iterations allowed in the convergence test
B	Z,T	Decision variable (similar to bonus multiple for zone)
BBOT	Z	Lower bound for B
BTOP	Z	Upper bound for B
BONDEL	—	Initial grid size for gradient calculations
MAXGTER	—	Maximum number of iterations searching for improvement within a single gradient calculation.
PLANYEARS	—	Number of years in the planning horizon
		Data for Inventory Projection
BETA	YETS,Z	Bonus coefficient in loss model equations
BETAE	Z	Bonus coefficient in extend-given-stay model equations
BETALB	Z	Coefficient on bonus value received at previous reenlistment
BONPAY	Z	Average basic pay for reenlistees in zone Z
BONCAP	—	Maximum amount of bonus offer (in dollars)
BONHIST	Z,H	Historical value of bonus multiple in the basic AFSC
CAREERS	BM	Flow into the AFSC in the CAREERS program for each bonus multiple (from the BEM)
CONTRT	J,YETS,YOS,C,T	Fraction of the inventory that will still be in the Air Force at T out of those in the cell a year earlier
FORCE	J,YETS,YOS,C,T	The initial inventory is read into the positions with T = 0
NPSTOE6		Proportion of YOS 0 cell that has a 6 year TOE
QUOTAL	T	Number of persons who will leave the SSL during the year that ends at T through the airman retraining program
SELRATE	J,T	Fraction of persons with grade J who will be promoted during year that ends at T

Table 10—continued

Name	Subscripts	Meaning
STSAMA	—	Fraction of career force entrants from the AFSC who will remain in the same occupation if no bonus is offered
STSAMB	—	Bonus coefficient in fraction of career force entrants from the AFSC who will remain in the same occupation if no bonus is offered
TOE6A	J,YETS,YOS,C	Probability that a person will choose a 6-year TOE given that he reenlists from cell J,YETS,YOS,C and does not receive a bonus
TOE6B	Z	Coefficient of the bonus incentive variable in the probability that a person will choose a 6-year TOE when some bonus is offered, given that he reenlists in zone Z
TOE6C	Z	Increase in the probability that a person will choose a 6-year TOE when some bonus is offered, given that he reenlists in zone Z

The penalty function that we wish to minimize is an infinite sum over all future years T = 1 to infinity:

$$\mathrm{OBJ} = \sum_{T} [(1-\mathrm{DISCOUNT})^{T}] \times \mathrm{OBJ(T)}. \tag{4.2}$$

For all T beyond the projection period, the program ensures that OBJT(T) is equal to SSOBJ, the annual penalty incurred by the steady-state target. Therefore, Eq. (4.2) is equivalent to:

$$\mathrm{OBJ} = \sum_{T=1}^{\mathrm{NTOP}} [(1-\mathrm{DISCOUNT})^{T}] \times \mathrm{OBJT(T)} + \\ [(1-\mathrm{DISCOUNT})^{\mathrm{NTOP}} / \mathrm{DISCOUNT}] \times \mathrm{SSOBJ}, \tag{4.3}$$

where NTOP is the number of years in the projection period.

The variables that control the optimization procedure, (part 2 of Table 10) are similar to the steady-state case, except that two new variables, BONDEL and MAXGTER, are used to control loops within the gradient calculation. The input data for the variable B(Z,T) should correspond to the steady-state value of SSB(Z) for all T.

Initial Inventory

The inventory at the end of T = 0 is stored in array FORCE by the initialization routine. This inventory includes all persons in the pipeline to enter the self-sustaining ladder (including a share of the lateral entrants in training, if any). It also includes persons who have been released through an early release program, but whose OETS was scheduled to occur after the start of the planning period (T = 0). Persons who reenlisted during the last year and whose OETS was scheduled for after T = 0 are counted as if they had not yet reenlisted.

Loss and Reenlistment Rates

Different continuation and reenlistment rates are stored for each year. For T = 1 to PLANYEARS + 4, the rates used for a projection will be calculated by the model. Although the decision variables only vary for T = 1 to PLANYEARS, the decision variables for zone A can affect rates four years into the future. The rates for later T will not be changed from the input.

For T = 1 to PLANYEARS + 4, the input continuation and loss rates are derived using the same assumptions as in the steady-state case—in particular, the assumption that bonuses are set at 0. For T > PLANYEARS + 4, the input continuation and reenlistment rates are based on the assumption that bonuses equal the value found in the steady-state solution (and thus the loss and reenlistment rates are identical to those found in the steady-state solution).

In the prototype, only PLANYEARS + 5 years of data are stored, since the array will be the same for all T ≥ PLANYEARS + 5. This results in a substantial savings of space but introduces a few complications in the calculations. In the interests of brevity and clarity, the specifications do not include this complication. In the current prototype, only a single year's rates are read in. Then differences due to economic conditions, if any, are read and the array for each year is filled in.

Airman Retraining Program

In addition to the CAREERS program, another airman retraining program flow into or out of the self-sustaining ladder may be simulated. The number leaving the AFSC during year T is input into QUOTAL(T). A negative value of QUOTAL signifies that this ladder is gaining personnel. The number should NOT include flows into or out of lateral entry AFSCs and should not include flows under the CAREERS program. The flow is distributed proportionally across cells in category of enlistment 2 or 3 with grade E-4 through E-7 and YOS ≤ 15.

Other Flows

The input data concerning promotions are identical to the steady-state input except for the fact that the select rate now has an annual dimension and therefore a new name.

The input variables STSAMA, STSAMB, and CAREERS are the same numbers as used for the steady-state case. The model assumes that the size of the CAREERS inflow does not change over time except in response to the decision variables B(1,T). The input variables governing TOE are also the same as those read in the steady-state model.

Initialization Calculations

Table 11 lists variables that are calculated during the initialization process (rather than just read from files). The authorizations for E-3 are adjusted to account for the inventory in training and are identical for each year beyond PLANYEARS. The projected inventory will include persons in training for the basic AFSC.[1] Rather than simulate the details of the actual pipeline for the AFSC, the model uses the size of the inventory in training found in the steady-state solution as an approximation of each year's inventory in training. The authorizations for grade E-3 are increased by that number. For each year T = 1 to PLANYEARS + 30, set:

$$A(3,T) = \text{authorizations for trained E-3s} + SSNPS \times STUDENTFR, \qquad (4.4)$$

where SSNPS is the number of YOS = 0 inventory in the steady-state solution, and STUDENTFR is the fraction of the YOS 0 inventory that is in training.

The entire term of enlistment choice array PTOE6D is initialized to the values used in the steady-state solution because the adjustment routine will calculate PTOE6D only during the planning horizon. The statements are similar to those in the steady-state routine except for the increase in dimensionality.

```
DO YOS = 2 TO 13
Z  = ZONE(YOS)
B = MIN(6,MAX(0,SSB(Z))
BONINC = MIN(BONCAP,6 × BONPAY(Z) × B) – MIN(BONCAP,4 × BONPAY(Z) × B)
IF SSB(Z) > 0 THEN SOMBON = 1
            ELSE SOMBON = 0
DO OVER C
```

[1] The model makes no attempt to account for persons in training for lateral entry specialties.

Table 11

Variables Calculated During Initialization Procedure

Name	Subscripts	Meaning
A	J,T	Only the E-3 authorizations are adjusted. The adjustment is given in Eq. (4.1)
ALPHAD	J,YETS,YOS,C,T	Fraction of the inventory that will still be in the Air Force a year later out of those in the cell at the beginning of the year, assuming bonus = 0 (duplicate of CONTRT array)
ALPHARD	J,YETS,YOS,C,T	Fraction that will reenlist during a year out of those who remain in the Air Force at the end of the year and who began the year in the cell, assuming bonus = 0 (duplicate of RENLGS array)
B	Z,T	Decision variable (similar to bonus multiple for zone). Set equal to BONHIST(Z,T) for T = – 5 to 0. Initialized to SSB(Z) for T ≥ 1
CARIN	T	Number of persons who will enter the AFSC through the CAREERS program. Initialized to CAREST, which is the number in the CAREERS program in the steady-state solution
ENDSTR	T	End strength (initialized to the sum of A(J,T) over all grades J). Includes allowance for inventory in training
PTOE6D	J,YETS,YOS,C,T	Probability that a person in the cell will choose a 6-year term of enlistment. Initialized to distribution from steady-state solution
ZONE	YOS	Zone associated with reaching OETS with YOS year of service. (ZONE (y) = 1 for y = 3, 4, and 5; ZONE (y) = 2 for y = 6 to 9, ZONE (y) = 3 for y = 10 to 13.)

```
DO OVER J
DO YETS = – 1 TO 1
DO T = 1 TO PLANYEARS + 30
PTOE6D(J,YETS,YOS,C,T) = TOE6A(J,YETS,YOS,C) + TOE6B(Z) × BONINC +
                         TOE6C(Z) × SOMBON
END END END END
```

Variables Received from the Steady State Solution

The variables that are passed from the steady-state optimizer to this program were discussed at the end of Sec. III. (See Table 8 for a list of these variables.)

ADJUSTING FLOW RATES

The subroutine DFLOW adjusts flow rates based on the values of the input array B(Z,T). The flow rates are adjusted using exactly the same logic as in the steady-state case.

Consequently, we just present the specifications and refer the reader to the subsection on adjusting the flow rates in Sec. III for the rationale. The variables that are adjusted in DFLOW are CONTRT, RENLGS, CARIN, and PTOE6D, which are described in either Table 10 or 11.

Continuation and Reenlistment Rates

The continuation rates and reenlistment rates are adjusted using the coefficients found in the latest fitting of the middle term loss models. Although the arrays in the specification contain rates for each possible projection year, only the rates for T = 1 to PLANYEARS + 4 are adjusted since the remaining rates never deviate from their steady-state values. A linear specification is recommended.[2] Because the code must be adapted to changes in the loss models, the following specification should be viewed as illustrative rather than definitive:

```
DO YOS = 3 TO 14
Z = ZONE(YOS)
DO T = 1 TO PLANYEARS + 4
DO OVER C
DO OVER J
CONTRT(J,1,YOS,C,T) = ALPHAD(J,1,YOS,C,T) - BETA(1,Z) × B(Z,T)
IF Z = 2 AND B(1,T - 4) > 0 THEN CONTRT(J,1,YOS,C,T) =
                                   CONTRT(J,1,YOS,C,T) + BETALB(Z)
IF Z = 1 THEN DO
    STAYSAME = STSAMA + STSAMB × MIN(6,MAX(0,B(Z,T)))
    RENLGS(J,1,YOS,C,T) = (ALPHARD(J,1,YOS,C,T) - BETAE(Z) × B(Z,T)) × STAYSAME
    RENLGS(J,0,YOS,C,T) = ALPHARD(J,0,YOS,C,T) × STAYSAME
    RENLGS(J, - 1,YOS,C,T) = ALPHARD(J, - 1,YOS,C,T) × STAYSAME
    END
ELSE RENLGS(J,1,YOS,C,T) = ALPHARD(J,1,YOS,C,T) - BETAE(Z) × B(Z,T)
END END END
```

CAREERS Flow Into Ladder

The array CARIN(T) gives the number of persons who will enter the ladder in the CAREERS program during year T. As in the steady-state case, linear interpolation is

[2] The current prototype uses a logit specification.

used to estimate the value for nonintegers between 1 and 6. The values of CARIN(T) for T > PLANYEARS are not adjusted and remain at their steady-state values.

```
DO T = 1 TO PLANYEARS
IF B(Z,T) ≤ 0 THEN CARIN(T) = CAREERS(0)
ELSE IF B(Z,T) ≥ 6 THEN CARIN(T) = CAREERS(6)
ELSE DO;
          BM = FLOOR(B(Z,T))
          CARIN(T) = CAREERS(BM) + (CAREERS(BM + 1) – CAREERS(BM)) ×
          (B(Z,T) – BM)
END END
```

Term of Enlistment Choice

The values of PTOE6D are adjusted for T = 1 to PLANYEARS using the following code (TB is a temporary variable to hold the value of the bonus that affects term of enlistment):

```
DO YOS = 3 TO 13
Z  = ZONE(YOS)
TB = MIN(6,MAX(0,B(Z,T))
BONINC = MIN(BONCAP,6 × BONPAY(Z) × TB) – MIN(BONCAP,4 × BONPAY(Z) × TB)
IF B(Z,T) > 0 THEN SOMBON = 1
            ELSE SOMBON = 0
DO OVER C
DO OVER J
DO YETS = – 1 TO 1
DO T = 1 TO PLANYEARS
PTOE6D(J,YETS,YOS,C,T) = TOE6A(J,YETS,YOS,C) + TOE6B(Z) × BONINC +
                        TOE6C(Z) × SOMBON
END END END END
```

PROJECTING THE INVENTORY

Figure 6 gives an overview of the calculations in the inventory projection subroutine. The projection for each year is completed before the projection for the next year is begun. Within each year the calculations proceed in order of increasing grade. Since the number of eligibles is known from the previous year's inventory, the calculation of promotion rates is simpler here than in the steady-state case. This model allows for an airmen retraining program that will move persons among SSLs. The distribution of each year's retraining flow among the IPM cells is calculated immediately before the calculation of NPS accessions for the year.

The flow within the IPM is controlled by two input variables, PROMGIVEN and PROJTOL. These and the other input variables used in the IPM are described in Table 12. When PROMGIVEN is 1, the promotion rates used in the projection are those found in array PRATE at the time of the call to the IPM. Otherwise, PRATE is calculated by the IPM. The calculations result in promotion rates for each year and grade equal to the input variable SELRATE(J,T) and a distribution of promotion rates across IPM cells proportional to the trial promotion rates. In the calls to the IPM from the iteration loop of the main program described in Fig. 5, PROMGIVEN is set to 0 and PRATE is calculated.

The first projection year is given by LOWYEAR, which is usually set to 1. When PROJTOL is 0, the projection period ends at the year input in the variable NYTOP. Otherwise, the projection proceeds until the inventory becomes close enough to the steady-state inventory that the penalty functions of the two inventories differ by less than PROJTOL or until T = NYTOP, whichever comes first. In the calls to the IPM from the iteration loop of the main program described in Fig. 5, PROJTOL is > 0 (10 has been used in the prototype runs) and NYTOP = PLANYEARS + 30.

The calculations are discussed below. The projection for each grade that would result from any given promotion rate is discussed first. Then we discuss the routines for calculating the promotion rate, the distribution of retrainees among IPM cells, and the number of NPS accessions. The last subsection covers the test for the end of the projection period. Table 13 describes the major variables calculated during the projection. The inventory projection model returns the arrays: FORCE, Y0, PROMD,RENLD, and

LATLOSS. The variable PRATE is both an input and an output array. The IPM also returns the number of years in the projection period in variable NTOPOUT.

Projecting the E-3 Inventory

The force of E-3s that continue into year T is calculated from the previous year's E-3 inventory using the following algorithm:

```
DO ITOE = 4 AND 6
DO YETS = – 1 TO ITOE – 1
YOS = ITOE – YETS
```

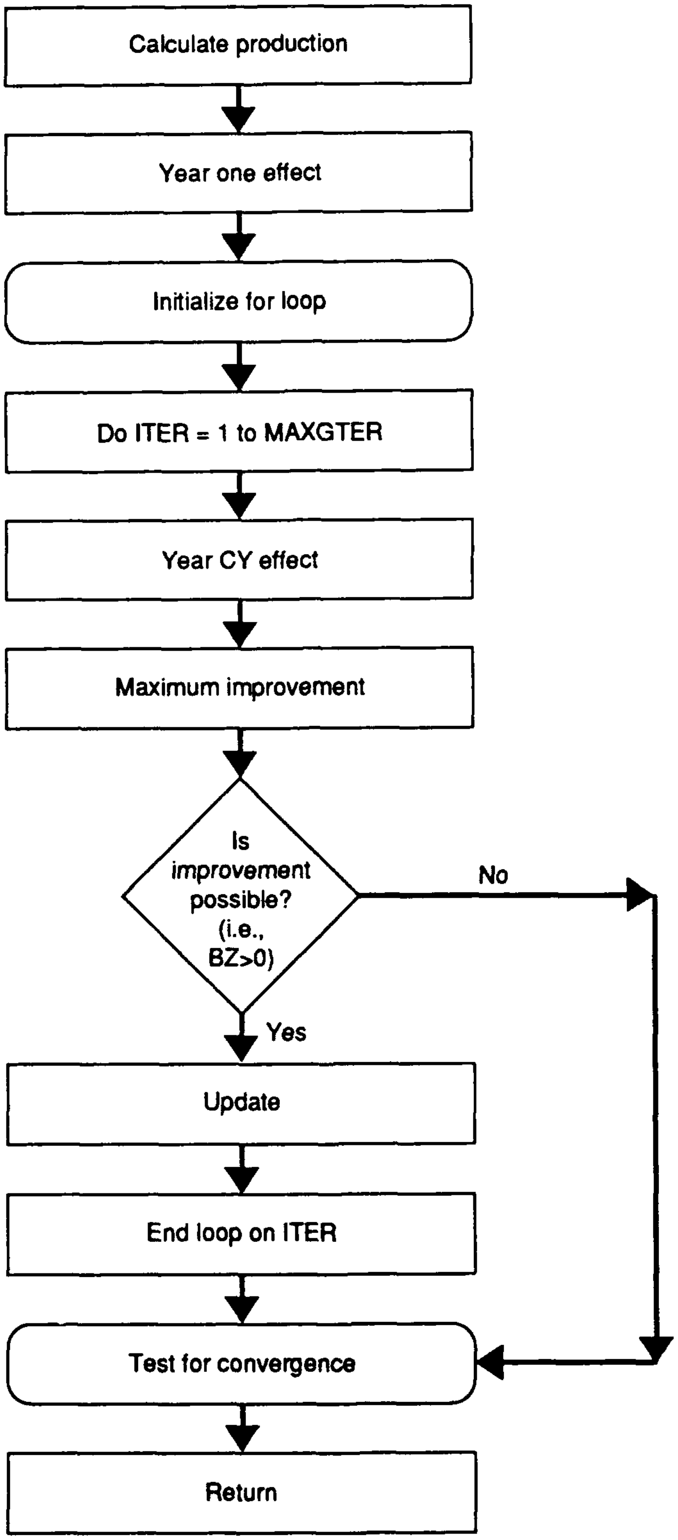

Fig. 6—Obtaining next values for B (z,t)

Table 12

Input Variables for Inventory Projection

Name	Subscripts	Meaning and Comments
CARDIST	J,YETS,YOS	Fraction of career force entries that join IPM cell. (The category of enlistment is 2 by definition). Used to distribute CAREERS retrainees among IPM cells
CARIN	T	Number of persons who will enter the AFSC through the CAREERS program
ENDSTR	T	End strength; includes allowance for inventory in training
CONTRT	J,YETS,YOS,C,T	Fraction of the inventory that will still be in the Air Force at T out of those in the cell a year earlier
LOWYEAR	—	First year of the projection period
NYTOP	—	Maximum number of years in the projection period
PROJTOL	—	Tolerance allowed in the difference between the penalty function in the last year of the projection and the steady-state value of the penalty function
PROMGIVEN	—	When PROMGIVEN = 1, the promotion rates per cell come from the input array PRATE. Otherwise, PRATE is calculated
PRATE	J,YOS,T	Fraction of those with grade J and YOS at end of T – 1 who will be promoted during year T. (Input values are used only when PROMGIVEN = 1)
PRATE3	YETS,YOS	Fraction of E-3 inventory in the YOS and YETS cell who will be promoted during the next year
PTRIAL	J,YOS	Initial (trial) value for the fraction of those with grade J and YOS who will be promoted in the next year. (Variable is 0 for grade E-3. Used only when PROMGIVEN ≠ 1)
RELNGS	J,YETS,YOS,C,T	Fraction that will reenlist during the year that ends at T out of those who remain in the Air Force at the end of the year and who began the year in the cell. (Note RENLGS(J,M,YOS,C,T) = 0 for all M > 1; RENLGS(J, – 1,YOS,C,T) = 1.0)
QUOTAL	T	Number of persons who will leave the SSL during the year that ends at T through the Airman Retraining Program (negative for gaining AFSCs)
SELRATE	J,T	Fraction of persons with grade J who will be promoted during year that ends at T. (Used only when PROMGIVEN ≠ 1)

```
FORCE(3,YETS,YOS,1,T) = FORCE(3,YETS + 1,YOS – 1,1,T – 1) × CONTRT(3,YETS +
                        1,YOS – 1,1,T) × (1 – RENLGS(3,YETS + 1,YOS – 1,1,T)) ×
                        (1 – PRATE3(YETS + 1,YOS – 1)).
PROMD(3,YETS,YOS,1,T) = FORCE(3,YETS + 1,YOS – 1,1,T – 1) × CONTRT(3,YETS +
                        1,YOS – 1,1,T) × (1 – RENLGS(3,YETS + 1,YOS – 1,1,T)) ×
                        PRATE3(YETS + 1,YOS – 1).
END END
```

Since E-3s aren't allowed to reenlist in the model, RENLD(3,YETS,YOS,C,T) is initialized to 0 for all cells. Also, PROMD(3,YETS,YOS,C,T) is initialized to 0 for all cells with C > 1.

Projecting Grades E-4 Through E-8

The equations are similar for grades E-4 through E-8 with the calculations for each preceding grade being completed before beginning the next grade.[3]

For category of enlistment 1, the only flows into the cell are promotions from the grade below and continuations of persons in the same grade who are neither promoted nor reenlisted.

```
DO ITOE = 4 AND 6
DO YETS = – 1 TO ITOE – 1
YOS = ITOE – YETS
FORCE(J,YETS,YOS,1,T) = PROMD(J – 1,YETS,YOS,1,T)
                      + FORCE(J,YETS + 1,YOS – 1,1,T – 1) × CONTRT(J,YETS +
                      1,YOS – 1,1,T) × (1 – RENLGS(J,YETS + 1,YOS – 1,1,T)) ×
                      (1-PRATE(J,YOS – 1,T))

PROMD(J,YETS,YOS,1,T) = FORCE(J,YETS + 1,YOS – 1,1,T – 1) × CONTRT(J,YETS +
                      1,YOS – 1,1,T) × (1 – RENLGS(J,YETS + 1,YOS – 1,1,T)) ×
                      PRATE(J,YOS – 1,T)
END END
```

[3] It is possible to eliminate some calculations by limiting the ranges through which the calculations are performed. For example, the first term does not contain persons in the highest grades. These savings are ignored, although some are actually achieved in the current prototype.

Table 13

Major Variables Calculated in Inventory Projection

Name	Subscripts	Meaning
FORCE	J,YETS,YOS,C,T	Inventory in cell
LATLOSS	J,YETS,YOS,C,T	The number of persons leaving the cell in the airman retraining program. (Negative for gaining AFSCs)
PROMD	J,YETS,YOS,C,T	Contains promotions out of grade J that will appear in the grade J + 1,YETS,YOS,C cell
RENLD	J,YETS,YOS,C,T	Contains reenlistments out of grade J that will appear in either the cell J + 1,YETS,YOS,C,T or in the cell J,YETS,YOS,C,T. RENL(3,YETS,YOS,C,T) is initialized to 0, and RENL(J,YETS,YOS,4,T) is never calculated
NTOPOUT	—	Number of years projected
PRATE	J,YOS,T	Fraction of those with grade J and YOS at end of T – 1 who will be promoted during year T
Y0	T	Number of persons in cell YOS = 0 in year T

For categories of enlistment 2 and 3, flows into the cells with YETS = 4 and YETS = 6 include reenlistments as well as promotions and continuations of those who are neither promoted nor reenlisted. The reenlistees can, of course, be promoted in the same year as they reenlisted. Reenlistments are calculated from:

$$\begin{aligned}\text{RENLD}(J,6,\text{YOS},C,T) = \text{SUM}\{&\text{FORCE}(J,m,\text{YOS}-1,c,T-1) \times \text{CONTRT}(J,m,\text{YOS}-1,c,T)\\ &\times \text{RENLGS}(J,m,\text{YOS}-1,c,T) \times \text{PTOE6D}(J,m,\text{YOS}-1,c,T)\}\end{aligned} \quad (4.5)$$

where the range of the summation depends on C. For C = 2, the sum is over m = – 1,0, and 1 and c = 1. For C = 3, the sum is over m = – 1,0, and 1 and c = 2 to 3. Similarly,

$$\begin{aligned}\text{RENLD}(J,4,\text{YOS},C,T) = \text{SUM}\{&\text{FORCE}(J,m,\text{YOS}-1,c,T-1) \times \text{CONTRT}(J,m,\text{YOS}-1,c,T)\\ &\times \text{RENLGS}(J,m,\text{YOS}-1,c,T) \times [1-\text{PTOE6D}(J,m,\text{YOS}-1,c,T)]\}.\end{aligned} \quad (4.6)$$

where the definition of the range is identical to that for the case of YETS = 6. With these calculated, the inventory can be projected as follows:

```
DO C = 2 TO 3
DO OVER YOS

FORCE(J,6,YOS,C,T) = RENLD(J,6,YOS,C,T) × (1 – PRATE(J,YOS – 1,T))
```

```
                    + PROMD(J – 1,6,YOS,C,T)
PROMD(J,6,YOS,C,T) = RENLD(J,6,YOS,C,T) × PRATE(J,YOS – 1,T)
FORCE(J,4,YOS,C,T) = RENLD(J,4,YOS,C,T) × (1 – PRATE(J,YOS – 1,T))
                    + PROMD(J – 1,4,YOS,C,T) + FORCE(J,5,YOS – 1,C,T – 1) ×
                    CONTRT(J,5,YOS – 1,C,T) × (1 – PRATE(J,YOS – 1,T))
PROMD(J,4,YOS,C,T) = RENLD(J,4,YOS,C,T) × PRATE(J,YOS – 1,T) +
             FORCE(J,5,YOS – 1,C,T – 1) × CONTRT(J,5,YOS – 1,C,T) ×
             PRATE(J,YOS – 1,T)

DO YETS = – 1 TO 3, 5
FORCE(J,YETS,YOS,C,T) = PROMD(J – 1,YETS,YOS,C,T)
                        + FORCE(J,YETS + 1,YOS – 1,C,T – 1) × CONTRT(J,YETS +
                        1,YOS – 1,C,T) × (1 – RENLGS(J,YETS + 1,YOS – 1,C,T)) ×
                        (1 – PRATE(J,YOS – 1,T))
        PROMD(J,YETS,YOS,C,T) = FORCE(J,YETS + 1,YOS – 1,C,T – 1)
                         × CONTRT(J,YETS + 1,YOS – 1,C,T) ×
                         (1 – RENLGS(J,YETS + 1,YOS – 1,C,T)) ×
                         PRATE(J,YOS – 1,T)
END END END
```

Category of enlistment 4 contains all persons with YOS greater than 19. The loss rates for years of service 19 and greater do not depend on YETS. For convenience, we store all retirement eligible persons in cells with YETS = 1 and reach the continuation rate for person with YOS greater than or equal to 19 from the cell with YETS = 1. We calculate the temporary variable R(J,T – 1) as the total of FORCE for year T – 1 summed over all cells with YOS = 19 and grade J.
Then:

```
FORCE(J,1,20,4,T) = PROMD(J – 1,1,20,4,T)
             + R(J,T – 1) × CONTRT(J,1,19,4,T) × (1 – PRATE(J,19,T))
PROMD(J,1,20,4,T) = R(J,T – 1) × CONTRT(J,1,19,4,T) × PRATE(J,19,T)

DO FOR YOS = 21 TO 29
```

```
FORCE(J,1,YOS,4,T) = PROMD(J – 1,1,YOS,4,T) + FORCE(J,1,YOS – 1,4,T – 1) ×
          CONTRT(J,1,YOS – 1,4,T) × (1 – PRATE(J,YOS – 1,T))
PROMD(J,1,YOS,4,T) = FORCE(J,1,YOS – 1,4,T – 1) × CONTRT(J,1,YOS – 1,4,T)
                              × PRATE(J,YOS – 1,T)
END
```

This concludes the description of the projection for any of the grades 4 through 8 for a given value of array PRATE.

Projecting the E-9 Inventory

The calculations for E-9 are simpler than for earlier grades, because there are no promotions out of E-9, and because their are no first term calculations. The calculations follow where J = 9:

```
DO C = 2 AND 3
DO OVER YOS

Calculate RENLD(J,6,YOS,C,T) FROM FORMULA 4.5
FORCE(J,6,YOS,C,T) = PROMD(J – 1,6,YOS,C,T) + RENLD(J,6,YOS,C,T)
FORCE(J,5,YOS,C,T) = PROMD(J – 1,5,YOS,C,T)
                                    + FORCE(J,6,YOS – 1,C,T – 1) × CONTRT(J,6,YOS – 1,C,T)

Calculate RENLD(J,4,YOS,C,T) FROM FORMULA 4.6
FORCE(J,4,YOS,C,T) = PROMD(J – 1,4,YOS,C,T) + RENLD(J,4,YOS,C,T)
                                    + FORCE(J,5,YOS – 1,C,T – 1) × CONTRT(J,5,YOS – 1,C,T)
DO YETS = – 1 to 3 and 5
FORCE(J,YETS,YOS,C,T) = PROMD(J – 1,YETS,YOS,C,T)
                         + FORCE(J,YETS + 1,YOS – 1,C,T – 1) × CONTRT(J,YETS + 1,YOS – 1,C,T)
                         × (1-RENLGS (J,YETS + 1,YOS – 1,C,T))
END END END
```

Then:

```
FORCE(J,1,20,4,T) = PROMD(J – 1,1,20,4,T)
```

```
                    + R(J,T – 1) × CONTRT(J,1,19,4,T)

DO FOR YOS = 21 TO 29
FORCE(J,1,YOS,4,T) = PROMD(J – 1,1,YOS,4,T)
                          + FORCE(J,1,YOS – 1,4,T – 1) × CONTRT(J,1,YOS – 1,4,T)
END
```

Determining Promotion Rate Parameters

This subsection describes the subroutines for determining PRATE(J,YOS,T), the promotion rates for each year of service that will produce an overall promotion rate equal to SELRATE(J,T). The subroutines process data for only one grade and year at a time. Two subroutines are involved. The function CALSEL calculates the promotion rates that were realized from the input trial promotion rates. GETNP calculates the PRATE values that will produce a promotion rate equal to SELRATE(J,T). These routines are called if, and only if, PROMGIVEN is not 1. The place where these routines are called is shown in Fig. 6.

The input to CALSEL consists of J, T, the results of the projection from the trial promotion rates for grade J and year T in array FORCE, and the promotion out of grade J in array PROMD. CALSEL calculates the temporary variables:

X = sum (FORCE(J,m,y,c,T – 1)), and
Y = sum(PROMD(J,m,y,c,T))

where the sums are over all YETS, YOS, and category of enlistments. The observed select rate, SR, is returned where:

SR = Y/X.

The subroutine GETNP calculates the values for PRATE(J,YOS,T) that will produce a promotion rate equal to SELRATE(J,T). The input consists of SELRATE(J,T), SR, and array PTRIAL. For each YOS, the new promotion rates are given by:

PRATE(J,YOS,T) = (SELRATE(J,T)/SR) × PTRIAL(J,YOS).

Spreading the Airman Retraining Program

There are two elements of the airman retraining program. First the retraining described in QUOTAL is spread to the IPM cell level. Then entrants to the AFSC through the CAREERS program are simulated.

The number of persons leaving each cell during year T is calculated and placed in array LATLOSS. The FORCE array is then adjusted. Let

$$X = \mathrm{SUM}(\mathrm{FORCE}(j,m,y,c,T))$$

be the sum of persons eligible for the airman retraining program. In the prototype this is the sum over $j = 4,5,6$, and 7, all values of YETS, all YOS ≤ 15, and $c = 2$ and 3. Then, for any cell within the same subscript ranges,

$$\mathrm{LATLOSS}(j,m,y,c,T) = (\mathrm{QUOTAL}(T)/X) \times \mathrm{FORCE}(j,m,y,c,T), \text{ and}$$
$$\mathrm{FORCE}(j,m,y,c,T) = \mathrm{FORCE}(j,m,y,c,T) - \mathrm{LATLOSS}(j,m,y,c,T).$$

The CAREERS program entrants increase the size of the YETS = 6 and YETS = 4 cells in the second category of enlistment according to the formula:

$$\mathrm{FORCE}(J,\mathrm{YETS},\mathrm{YOS},2,T) = \mathrm{FORCE}(J,\mathrm{YETS},\mathrm{YOS},2,T) + \mathrm{CARIN}(T) \times \mathrm{CARDIST}(J,\mathrm{YETS},\mathrm{YOS}).$$

Calculating the YOS = 0 Cells

Finally, the number of persons present at the end of the fiscal year T from accessions during T is calculated to meet the total desired end strength. (Note that this includes an inventory in training equal to the steady-state inventory in training.) The total desired number of persons in the YOS 0 cells is:

$$\mathrm{Y0}(T) = \mathrm{MAX}(0.5, \mathrm{ENDSTR}(T) - \Sigma\, \mathrm{FORCE}(j,m,y,c,T)),$$

where the sum is over all grades, YETS, categories of enlistment, and all YOS values except 0. The distribution is:

$$\mathrm{FORCE}(3,6,0,1,T) = \mathrm{Y0}(T) \times \mathrm{NPSTOE6}(T)$$
$$\mathrm{FORCE}(3,4,0,1,T) = \mathrm{Y0}(T) \times (1 - \mathrm{NPSTOE6}(T)).$$

Testing for End of Projection Period

When T > PLANYEARS + 4 and PROJTOL > 0 and T < NYTOP, a test is performed to see whether the projection may reasonably end. The projection will end when the penalty function for the year is within PROJTOL of the steady-state penalty function value. First calculate the inventory in each grade, FBYGD(J,T),

$$\mathrm{FBYGD}(J,T) = \text{sum } \mathrm{FORCE}(J,m,y,c,T).$$

Then calculate the year's penalty function:

```
$OBJ = 0$
DO $J = 3$ TO 9
$OBJ = OBJ + W(J) \times (A(J,T) - FBYGD(J,T))^2$
END
```

IF ABS(OBJ – SSOBJ) is less then PROJTOL, or T = NYTOP, then set NTOPOUT equal to T and return control to the calling program. Otherwise set T = T + 1 and project for another year.

THE INITIAL SOLUTION

This section describes the calculation of the first nonsteady-state position of the decision variables. The aim of the calculation is, for each planning year, to have the numbers of persons staying in the Air Force beyond their ETS date in each zone be proportional to the same numbers in the steady-state solution. The number of NPS accessions will be in the same ratio to the steady-state NPS accessions as the continuations are to the steady-state continuations. Persons who reach their ETS are called "decisionmakers."

To obtain a starting value for the number of decisionmakers, a preliminary inventory projection for PLANYEARS into the future is performed with the IPM flow parameters set equal to the steady-state values; in particular CONTRT contains the continuation rates at the steady-state bonus values.[4] This produces the steady-state ETS continuation rates, but the initial solution is to have ETS continuation counts close to the steady-state ETS continuation counts.

[4] The arguments to the IPM for this first projection have PROMGIVEN = 0, NYTOP = PLANYEARS + 1, PROJTOL = 0, and all flow rates equal to their steady-state values.

We store the number of persons continuing past ETS in this preliminary projection in the temporary array CURCONT(Z,T) and the number of decisionmakers in the variable DM(Z,T). After we initialize these 2 arrays to zero, these variables are calculated as follows:

```
DO T = 1 TO PLANYEARS
  CURCONT(1,T) = CARIN(T)
  DO YOS = 3 TO 13;
    Z = ZONE(YOS)
    DO OVER C
      DO OVER J
        CURCONT(Z,T)=CURCONT(Z,T) + FORCE(J,1,YOS,C,T – 1) × CONTRT(J,1,YOS,C,T)
        DM(Z,T)= DM(Z,T) + FORCE(J,1,YOS,C,T – 1)
END END END END
```

Let IPCONT(Z,T) be the desired number of persons continuing past ETS in zone Z, year T, and IPNPS(T) be the desired NPS accessions. By definition, there is a number C(T) so that:

$$\mathrm{IPCONT}(Z,T)=C(T)\times \mathrm{SSCONT}(Z),$$

$$\mathrm{IPNPS}(T)=C(T)\times \mathrm{SSNPS}, \text{and}$$

$$\mathrm{IPNPS}(T)+\sum_{Z}\mathrm{IPCONT}(Z,T)=\mathrm{YO}(T)+\sum_{Z}\mathrm{CURCONT}(Z,T).$$

The solution is:

$$C(T)=\frac{\mathrm{YO}(T)+\sum\limits_{Z}\mathrm{CURCONT}(Z,T)}{\mathrm{SSNPS}+\sum\limits_{Z}\mathrm{SSCONT}(Z)}$$

After calculating the CURCONT array, the algorithm calculates C(T) and then IPCONT and IPNPS. The remaining problem is to go from the values of IPCONT to the desired values of B. This is straightforward for Z = 2 and 3:

$$B(Z,T)=\mathrm{SSB}(Z)+\frac{(\mathrm{IPCONT}(Z,T)-\mathrm{CURCONT}(Z,T)}{\mathrm{DM}(Z,T)\times \mathrm{BETA}(1,Z)}$$

For Z = 1, we need to account for the CAREERS program flow also. We first obtain the constant term of the first term ETS continuation rate averaged over all IPM cells and store it in temporary variable AVGALPHA. Then we test for the interval in the CAREERS array that will produce the desired continuations. The algorithm is:

```
DO T = 1 TO PLANYEARS
AVGALPHA = (CURCONT(1,T)CARIN(T))/DM(1,T) – BETA(1,1) × SSB(1)
IF IPCONT(1,T) < AVGALPHA × DM(1,T) + CAREERS(0) THEN B(1,T) =
          ((IPCONT(1,T) – CAREERS(0))/DM(1,T) – AVGALPHA)/BETA(1,1)
ELSE DO
    NOTDONE = 1
    DO BM = 1 TO 6 WHILE NOTDONE
    IF IPMCONT(1,T) < (AVGALPHA + BETA(1,1) × BM) × DM(1,T) + CAREERS(BM)
    THEN
            DO
            NOTDONE = 0
            B(1,T) = BM – 1 +
(IPCONT(1,T) – CAREERS(BM – 1) – DM(1,T) × (AVGALPHA + BETA(1,1) × (BM – 1)))/
      (CAREERS(BM) – CAREERS(BM – 1) + BETA(1,1) × DM(1,T))
    END END
    IF NOTDONE = 1 THEN B(1,T) =
          ((IPMCONT(1,T) – CAREERS(6))/DM(1,T) – AVGALPHA)/BETA(1,1)
    END END
```

The final step in the determining the initial position is to insure that the chosen values of the decision variables are within the input constraints:

```
DO T = 1 TO PLANYEARS
DO Z = 1 TO 3
IF B(Z,T) < BBOT(Z) THEN B(Z,T) = BBOT(Z)
IF B(Z,T) > BTOP(Z) THEN B(Z,T) = BTOP(Z)
END END
```

CHOOSING IMPROVED VALUE OF BONUS VARIABLE

This subsection describes the algorithm that controls the optimizing calculations. An approximation to the gradient is calculated and then a search strategy is used to find an improved value of B. That improved value for the decision variable array is returned to the calling program in array BONNEXT. This subsection also describes the convergence test of whether the suggested changes in the decision variables are small enough to indicate that convergence has occurred.

Table 14 describes the major variables that are used in these calculations. The approximation to the gradient is based on an estimate of the number of persons who will be present and in grade J at the end of some year, say T, out of the number who reach OETS in an earlier year, say CY. Thus a second time subscript is needed, and we will use CY to refer to the year containing a particular cohort's OETS.

Figure 7 breaks the algorithm down into several steps. First it determines which fraction of each year's inventory by grade were decisionmakers during the planning horizon. Those who remain in the inventory are the production from the zone and cohort year, and they are counted in array PRODB. The production subroutine also counts the number of

Table 14

Major Local Variables Used in Calculating the Next Value of B

Name	Subscripts	Meaning
AROBJ	Z,CY	Estimated penalty function when BONNEXT(Z,T) changes from current value to current value + BONDEL
BONNEXT	Z,CY	Suggested next value for the decision variables
DM	Z,CY	Number of persons reaching OETS in zone Z during year CY
FBYGD	J,T	Inventory in grade J at end of year T
ERENL	Z,CY	Number of persons who will ever reenlist out of those who reach OETS in zone Z during year CY
PRODB	J,Z,T,CY	Number of persons who reach OETS in zone Z during year CY who will be present and in grade J at the end of year T at current values for B(Z,T) (the production of J,T from Z,CY)
RENLEFF	J,Z,T	Amount by which ERENL(Z,1) increases when B(Z,1) goes from current value to current value + BONDEL expressed as a fraction of DM(Z,1); excludes changes in the CAREERS program)
NPSEFF	J,Z,T	Increase in production of grade J at end of year T from NPS accessions after T ≥ 0 that will result from an increase in the bonus B(Z,1) from current value to current value + BONDEL (expressed as a fraction of DM(Z,1)

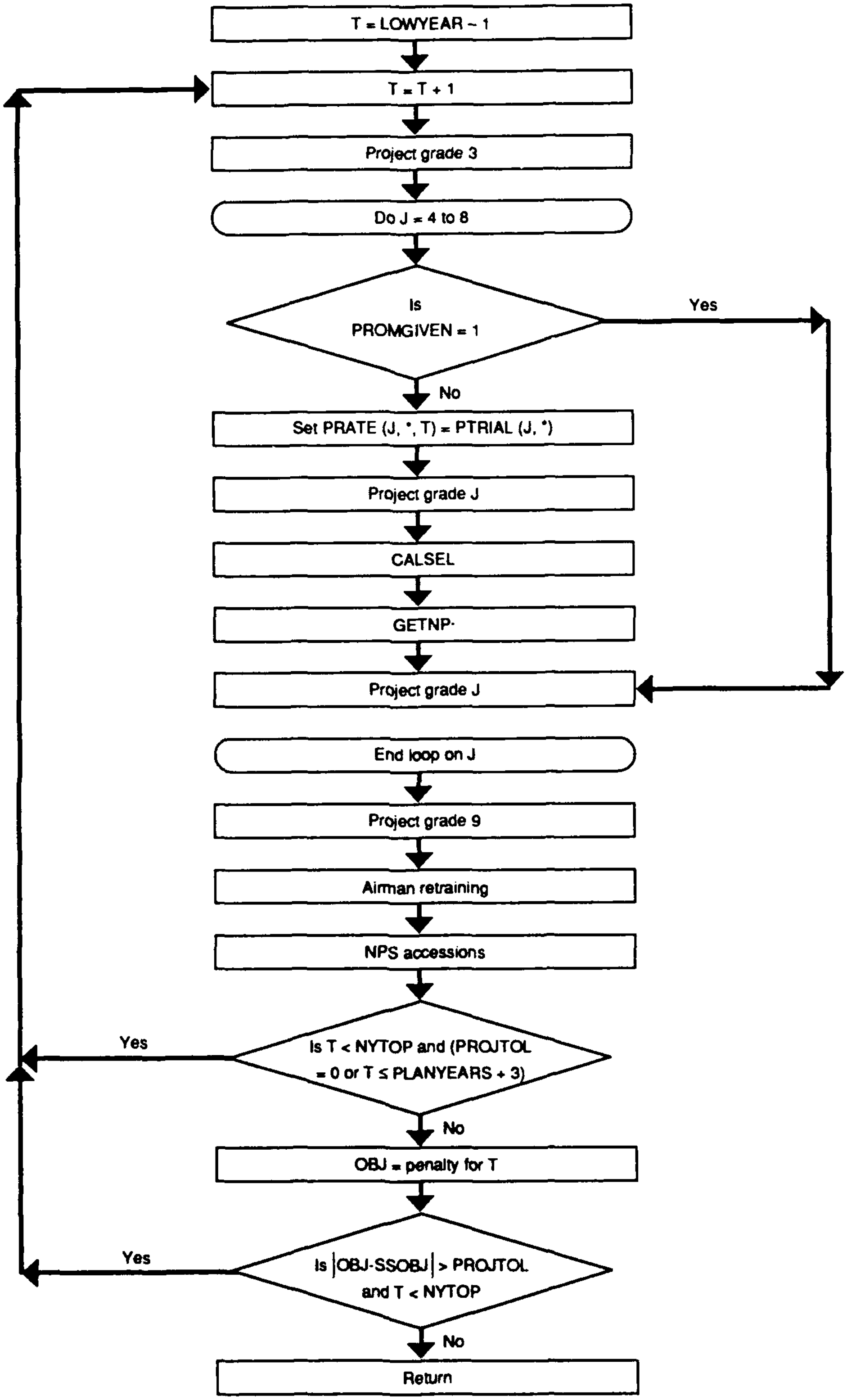

Fig. 7—Inventory projection

decisionmakers and the number of those who will ever reenlist. The effects of an increase in B(Z,T) for T = 1 on the number of persons who will ever reenlist is calculated in the "Year one effect" subroutine. After initializing BONNEXT to the current value of B, an iterative loop is entered that estimates the the effect of an increase of size BONDEL in BONNEXT(Z,CY) for each zone and cohort year. The zone and cohort that produce the maximum decrease in the penalty function are selected, and the effect of this decision on inventory counts by grade and year is then estimated in the update subroutine. The loop terminates when there is no improvement or when the loop has been performed MAXGTER times.

Production Calculations

First calculate (1) DM(Z,CY), the number of persons who will reach OETS in zone Z during cohort year CY; (2) ERENL(Z,CY), the number of persons from the zone and cohort who will ever reenlist; and (3) PRODB(J,Z,T,CY), the number of persons from the zone and cohort who will be in grade J at the end of year T. Because persons reenlist at different times, the cohorts are mingled within IPM cells. Consequently, the IPM is run for each cohort year, CY.

Each call to the IPM projects from year CY to NTOPOUT, the number of years projected in the last call to the IPM from the main program. The calls to the IPM use duplicate variables for FORCE (DUPFOR), RENLD(DUPRENL), and promotions to save the current values for later use. A duplicate of the input argument to the IPM QUOTAL(DUPQUOT) is set to 0 as this retraining flow is not affected by the decision variables. A duplicate of the CARIN array (DUPCAR) is set to 0 for all years except CY. The algorithm follows:

```
NYTOP = NTOPOUT
PROJTOL = 0
PROMGIVEN = 1
DUPQUOT = 0
PRODB = 0
DM = 0
ERENL = 0
DO CY = 1 TO PLANYEARS
DUPFOR = 0
DO YOS = 3 TO 13
Z = ZONE(YOS)
```

```
DO OVER J
DO OVER C
DM(Z,CY) = DM(Z,CY) + FORCE(J,1,YOS,C,CY – 1)
DUPFOR(J,1,YOS,C,CY – 1) = FORCE(J,1,YOS,C,CY – 1)
END END END
DUPCAR = 0
DUPCAR(CY) = CARIN(CY)
LOWYEAR = CY
CALL IPM
DO T = CY TO NYTOP
DO Y = 3 TO 13
Z = ZONE(Y)
YOS = Y + T + 1 – CY
IF YOS ≤ 30 THEN DO OVER J
DO OVER C
DO OVER YETS
PRODB(J,Z,T – CY+1,CY) = PRODB(J,Z,T – CY + 1,CY) + DUPFOR(J,YETS,YOS,C,T)
IF T ≤ CY + 2 THEN ERENL(Z,CY) = ERENL(Z,CY) + DUPRENL(J,YETS,YOS,C,T)
END END END END END

DO CY = 1 TO PLANYEARS
ERENL(1,CY) = ERENL(1,CY) – CARIN(CY)
END
```

Year One Effect

We next estimate the increased production of inventory in each grade and year that will result from an increase in the bonus offered at T = 1 in each zone. The amount of the bonus increase is BONDEL. The net increase in production is the sum of two parts, an increase in the cohorts who passed OETS during the first year and a decrease in the number of persons produced by accessions during year 1 and subsequent years.

The increase in production is estimated by a proportional relationship to the increase in the number of persons who ever reenlist. The number who reenlist in response to the increased year one bonus is stored in zone Z, excluding the input from the CAREERS program, in array EXRENL(Z), and then the increase in reenlistments is expressed as a fraction of the number of decisionmakers in array RENLEFF(Z). The decrease in the number of persons produced by accessions is accumulated in array NPSEFF, which is then normalized by dividing by the number of decisionmakers.

First initialize the IPM parameters. For each zone, set the flow rates to those associated with the increased bonus and call the IPM. These calls to the IPM begin the projection period at year 1. They pass the values of CARIN associated with the bonus plan and the user input values for the airmen retraining array QUOTAL. All other arguments to the IPM are the same as those used above.

```
LOWYEAR = 1
NYTOP = NTOPOUT
PROJTOL = 0
PROMGIVEN = 1
DUPFOR = 0
DO OVER YOS
DO OVER YETS
DO OVER J
DO OVER C
DUPFOR(J,YETS,YOS,C,0) = FORCE(J,YETS,YOS,C,0)
END END END END
NPSEFF = 0
DO Z = 1 TO 3
B(Z,1) = B(Z,1) + BONDEL
CALL DFLOW
CALL IPM
B(Z,1) = B(Z,1) – BONDEL
EXRENL(Z) = 0
DO OVER YOS for zone Z
DO OVER J
DO OVER C
EXRENL(Z) = EXRENL(Z) +
        DUPFOR(J,1,YOS,C,0) × CONTRT(J,1,YOS,C,1) × RENLGS(J,1,YOS,C,1) +
        DUPFOR(J,0,YOS + 1,C,1) × CONTRT(J,0,YOS + 1,C,2) × RENLGS(J,1,YOS +
        1,C,2) + DUPFOR(J, – 1,YOS + 2,C,2) × CONTRT(J, – 1,YOS + 2,C,3)
END END END
RENLEFF(Z) = (EXRENL(Z) – ERENL(Z,1))/DM(Z,1)
```

```
DO T = 1 TO NYTOP
  DO OVER J
    DO YOS = 0 TO T - 1
      DO OVER C
        DO OVER YETS
          NPSEFF(J,Z,T) = NPSEFF(J,Z,T) + DUPFOR(J,YETS,YOS,C,T) - FORCE(J,YETS,YOS,C,T)
        END END END
    NPSEFF(J,Z,T) = NPSEFF(J,Z,T)/DM(Z,1)
END END END
```

Calculating the Penalty Function

A function, which will be called PENALTY, accepts as input projected counts of the force in each year and grade, FBYGD, and returns V, the value of the penalty function for that projected inventory. This function will be used to estimate how the penalty function will vary with changes in B(Z,T) for each Z and T. The specification of the function follows:

```
V = 0
DCT = 1
DO T = 1 TO NTOPOUT
  OBJ = 0
  DO J = 3 TO 9
    $OBJ = OBJ + W(J) \times (A(J,T) - FBYGD(J,T))^2$
  END
  $DCT = DCT \times (1 - DISCOUNT)$
  $V = V + DCT \times OBJ$
END
$V = V + DCT \times SSOBJ/DISCOUNT$
RETURN
```

Initialize for Iterative Loop

Each pass through the iterative loop shown in Fig. 7 results in either an increase or a decrease in one item of the array BONNEXT(Z,T) by the amount BONDEL. An estimate of the inventory by grade and year that would result from the "bonus plan" contained in BONNEXT is maintained in the array FBYGD; and an estimate of the value of the penalty function that would result from the same "bonus plan" is maintained in variable X. The initialization routine merely initializes these three variables.

For Z = 1 to 3 and CY = 1 to PLANYEARS, set:

$$\mathrm{BONNEXT}(Z,CY) = B(Z,CY).$$

For J = 3 to 9 and T = 0 to NTOPOUT set:

$$\mathrm{FBYGD}(J,T) = \Sigma\,(\mathrm{FORCE}(J,m,y,c,T)),$$

where the sum is over all YETS, YOS, and categories of enlistment. And set:

$$X = \mathrm{PENALTY}(\mathrm{FBYGD}).$$

Year T Effect Approximation

In this subroutine estimate, for each Z and CY, the penalty function that would result if the decision variable were increased from BONNEXT(Z,CY) to BONNEXT(Z,CY) + BONDEL. This is an approximation based on the rationale that (1) the inventory produced by a bonus will be proportional to the number of the cohort who ever reenlist, and (2) the increase in the number of the cohort who ever reenlist will be proportional to those reaching OETS (except for CAREERS effects).[5] Thus the net increase in the number of persons in grade J in year T is estimated in response to an increase of size BONDEL in B(Z,CY) to be given by:

$$\begin{aligned} Y = {} & \mathrm{PRODB}(J,Z,T,CY) \times [\mathrm{RENLEFF}(Z) \times \mathrm{DM}(Z,CY) + \mathrm{CARDEL}]/[\mathrm{ERENL}(Z,CY) + \\ & \mathrm{CARIN}(IY)] + \mathrm{NPSEFF}(J,Z,T) \times \mathrm{DM}(Z,CY) \end{aligned} \tag{4.7}$$

where CARDEL is the increase in the size of the CAREERS program. (CARDEL is 0 for Z > 1.)

The estimate of the penalty function is stored in array AROBJ. The calculations of the change in the penalty function involve a duplicate copy of array FBYGD, which is updated for

[5] This approximation is slightly different in the prototype because the prototype has logit loss rates,and the CAREERS program is handled differently in the prototype.

$T \geq CY$ using Eq. (4.7). (Naturally, changing the decision variables for year CY does not change the inventory in the years preceding CY.)

```
AROBJ = 0
DO CY = 1 TO PLANYEARS
DO J = 3 TO 9
TBYGD(J,CY – 1) = FBYGD(J,CY – 1)
END
DO Z = 1 TO 3
CARDEL = 0
IF Z = 1 AND BONNEXT(Z,CY) ≥ 0 AND BONNEXT(C,CY) + BONDEL < 6 THEN DO
                  BM = FLOOR(BONNEXT(Z,CY))
                  CARDEL = (CAREERS(BM + 1) – CAREERS(BM)) × BONDEL
                  END
DO T = CY TO NTOPOUT
DO J = 3 TO 9
IF Z = 1 THEN DENOM = ERENL(Z) + CARIN(CY)
         ELSE DENOM = ERENL(Z)
TBYGD(J,T) = FBYDG(J,T) +
        PRODB(J,Z,T,CY) × (RENLEFF(Z) × DM(Z,CY) + CARDEL)/DENOM +
            NPSEFF(J,Z,T) × DM(Z,CY)
END END
AROBJ(Z,CY) = PENALTY(TBYGD)
END END
```

Maximum Improvement

In this subroutine, search array AROBJ to determine the zone and cohort such that a change in the decision variable of size BONDEL will produce the largest improvement in the penalty function. The change can be either an increase or a decrease in the decision variable. The subroutine returns the chosen zone and cohort in variables BZ and BCY. It also returns variable S, which gives the direction of the change. S is set to 1 if the decision variable should increase and to – 1 otherwise.

The other input required for the subroutine is X, the current value of the penalty function. The minimum value of the penalty function found at any time in the search is stored in MINX. The subroutine ensures that the chosen decision variable will be within the input constraints in arrays BBOT and BTOP.

For each value of zone and cohort, the algorithm checks if an increase in the bonus will result in a smaller value of the penalty function (if AROBJ(Z < CY) > MINX) and that an increase in the bonus is allowed by the constraint BTOP(Z,CY). If so, this is taken as a provisional solution. Similarly, the program checks if a decrease in the bonus will result in a smaller value of the penalty function. A decrease in the bonus will result in a penalty function value of X – (AROBJ(Z,CY) – X).

It may happen that no improvement is possible because all of the decision variables are up against their constraints. In that case the variable BZ will be returned at the value 0. As shown in Fig. 7, this outcome terminates the iterative loop.

```
MINX = X
BZ = 0
BCY = 0
S = 0
DO Z = 1 TO 3
DO CY = 1 TO PLANYEARS
IF AROBJ(Z,CY) < MINX AND BONNEXT(Z,CY) + BONDEL ≤ BTOP(Z,CY) THEN DO
        BZ = Z
        BCY = CY
        S = 1
        MINX = AROBJ(Z,CY)
        END
ELSE IF AROBJ(Z,CY) > X AND X – (AROBJ(Z,CY) – X) < MINX
                         AND BONNEXT(Z,CY) – BONDEL ≥ BBOT(Z,CY) THEN DO
      BZ = Z
      BCY = CY
      S = – 1
      MINX = X – (AROBJ(Z,CY) – X)
      END END END
```

Update Routine

The update routine takes the findings of the previous subroutine and updates all the variables for the next iteration. The variables to be adjusted are BONNEXT, FBYGD, and X.

The calculation of FBYGD again involves the approximation of Eq. (4.7), but this time the direction of the change is determined by S.

```
BONNEXT(BZ,BCY) = BONNEXT(BZ,BCY) + S × BONDEL
CARDEL = 0
IF BZ = 1 AND ((S = 1 AND BONNEXT(BZ,BCY) ≥ 0 AND
            BONNEXT(BZ,BCY) + BONDEL < 6) OR
            (S = – 1 AND BONNEXT(BZ,BCY) – BONDEL ≥ 0
            AND BONNEXT (BZ,BCY) ≤ 6)) THEN D0
                  BM = FLOOR(BONNEXT(BZ,BCY))
                  CARDEL = (CAREERS(BM + 1) – CAREERS(BM)) × BONDEL
                  END
CARINIT = 0
IF BZ = 1 THEN CARINIT = CARIN(BCY)
DO T = BCY TO NTOPOUT
DO J = 3 TO 9
FBYGD(J,T) = FBYDG(J,T) + S
        × (PRODB(J,BZ,T,BCY) × (RENLEFF(BZ) × DM(BZ,BCY) + CARDEL)/
        (ERENL(BZ,BCY) + CARINIT) + NPSEFF(J,BZ,T) × DM(BZ,BCY))
END END
X = PENALTY(FBYGD)
```

Convergence Test

The loop whose elements have just been described terminates either because it has been performed MAXGTER times or because all the decision variables are constrained. After the loop terminates, the remaining activity is to determine whether the suggested changes in the decision variables are small enough to indicate convergence of the entire dynamic optimizing program described in Fig. 5. The check is whether the difference in any variable exceeds the input variable STEPTOL.

```
ICONV = 1
DO Z = 1 TO 3
DO T = 1 TO PLANYEARS
```

```
IF ABS(B(Z,T) – BONNEXT(Z,T)) > STEPTOL THEN ICONV = 0
END END
```

If ICONV remains = 1 at the end of the loop, or if the main loop has been processed more than MAXLOOP times, the calculations for this self-sustaining ladder are complete and the solution is written out. Otherwise, the array B is set equal to BONNEXT over the planning horizon, and another iteration occurs beginning with a call to DFLOW.

OUTPUT

The primary outputs from the dynamic optimizer are the annual targets contained in array FORCE. These can be input to the AFSC target allocation module or directly to the target analysis module.

Another variable of interest is the number of reenlistments in the target for each year. The number of persons reenlisting into each cell can be calculated in a manner that is analogous to that used for the steady-state model:

$$\begin{aligned}&\mathrm{RENLIND(J,YETS,YOS,C,T)} =\\ &\qquad \mathrm{RENLD(J,YETS,YOS,C,T)} \times (1 - \mathrm{PRATE(J,YOS} - 1,\mathrm{T)} +\\ &\qquad \mathrm{RENLD(J} - 1,\mathrm{YETS,YOS,C,T)} \times \mathrm{PRATE(J} - 1,\mathrm{YOS} - 1,\mathrm{T)}.\end{aligned}$$

These reenlistment counts do not include retrainees entering through the CAREERS program. To add these to obtain total reenlistments into the second term, use the following formula:

$$\begin{aligned}&\mathrm{RENLIND(J,YETS,YOS,2,T)} = \mathrm{RENLIND(J,YETS,YOS,2,T)} +\\ &\qquad\qquad \mathrm{CARIN(T)} \times \mathrm{CARDIST(J,YETS,YOS)}.\end{aligned}$$

These reenlistment counts may then be aggregated in any manner desired.

The user might also wish to see other variables either on a paper printout or displayed within the target analysis module. Of particular interest are the values of the decision variables (B(Z,T)), the number of NPS accessions each year (Y0(T)), the number of entrants in the careers program (CARIN(T)), and the number of decisionmakers who reach ETS in the zone. The latter is given by:

$$DM(Z,T) = sum(FORCE(j,1,y,c,T)),$$

where the sum is over all grades j and categories of enlistment and the values of YOS that are appropriate to the zone.

V. ALLOCATING TARGETS TO AFSCs

The dynamic optimizer writes out a file containing a target force for each year for each self-sustaining ladder. The steady-state optimizer writes out a similar file with a single target force for each year in the distant future. This section provides the algorithm that transforms the set of targets for SSLs into a set of targets for AFSCs. The process is repeated identically for each year for which targets are sought.[1] The same process can be applied to the set of steady-state targets for SSLs to obtain long-run targets for AFSCs.

The process of attributing SSL targets to AFSCs is essentially the inverse of the process, described in Sec. II, that creates authorizations for SSLs from authorizations for AFSCs. This process begins by allocating each ladder's target among the basic AFSC and each of the laterals that it supports. Then it allocates the target to cappers. These descriptions assume that targets are available for all the SSLs and that targets are wanted for all the AFSCs. The same process may be used if only one or a small number of SSLs is studied, but targets for laterals and cappers that are supported in part by excluded SSLs should be discarded.

The subscripts used in this section have all been defined previously but are repeated in Table 15 for convenience. For simplicity, the subscript for year is discarded on all the variables as the calculations are repeated identically for each year.

The input data are listed in Table 16. For each SSL, the input target, LTARG, is either one year's value of the FORCE array from dynamic or the entire array SSF from the steady-state optimizer. The authorization counts for laterals, and the lateral-basic-feeder

Table 15

Subscripts for AFSC Target Allocation

Name	Range	Meaning
C	1 to 4	Category of enlistment
J	3 to 9	Grade
YETS	- 1 to 6	Years before ETS
YOS	0 to 29	Years of service
I	alphabetic	AFSC
IL	alphabetic	Lateral AFSCs
INDEX	1 – 99	Position of feeder in lateral-basic-feeder table

[1] The prototype is programmed to process three years simultaneously.

Table 16

Input Variables to Target Allocation Program

Name	Subscripts	Meaning
LTARG	J,YOS,YETS,C,I	Target for SSL I
FAF	IL, INDEX	List of SSLs that feed lateral IL (from the lateral-basic-feeder table)
X	IL, INDEX	Fraction of entrants to lateral IL that come from SSL FAF(IL,INDEX) (from the lateral basic-feeder table)
AUTHC	IL,J	Authorizations for lateral AFSCs, including their share of capper authorizations

table arrays, were constructed when the authorizations were grouped into SSLs as described in Sec. II.

LATERAL TARGETS

This section divides the input targets for SSLs into (1) targets for each basic AFSC and its cappers and (2) targets for each lateral AFSC and its cappers. The division between noncapper AFSCs and AFSCs is described below. The output is placed in the array TARGET, which has the same set of subscripts as does the disaggregate IPM inventory: grade J, YETS, YOS, category of enlistment C, and AFSC I. The array is initialized to 0 in all cells.

As described in Sec. II, the lateral-basic-feeder table consists of 2 two-dimensional arrays. For each value of INDEX (1) FAF(IL, INDEX) contains the name of a basic AFSC that provides entrants to lateral IL, either directly or indirectly through other laterals and/or cappers and (2) X(IL,INDEX) contains the fraction of entrants to lateral IL that comes either directly or indirectly from FAF(IL,INDEX). The authorizations for each IL were then divided among SSLs so that the SSL named FAF(IL,INDEX)[2] contains the fraction X(IL,INDEX) of the authorizations for IL in each grade.

This algorithm reverses this process. The same array X is used to allocate the target for each SSL in such a way that the proportion of the target for each lateral IL that comes from the SSL with the name FAF(IL,INDEX) is given by X(IL,INDEX). Thus the total number of positions in the target for grade J and lateral IL that come from the SSL

[2] Recall that the list of basic AFSCs coincides with the list of SSLs.

FAF(IL,INDEX) is given by AUTHC(IL,J).× X(IL,INDEX). That is, for all J and lateral AFSCs IL,

$$\text{sum[TARGET(J,m,y,c,IL)]} = \text{AUTHC(IL,J)} \times \text{X(IL,index)}, \tag{5.1}$$

where the sum is over all values of YETS, YOS, and categories of enlistment. The program begins by calculating the fraction of the input target from SSL I, which should be allocated to lateral IL. This fraction depends on grade and category of enlistment. For all grades except grade 3, only the career force portion of the target ($C \geq 2$) is allocated to laterals. This decision is consistent with the fact that almost no lateral entrants are drawn from the first term force. Few laterals contain authorizations for grade 3. To accommodate those that do, we allow a lateral target to contain a first term portion for grade 3.

Define SUPFR(J,C,I,IL) to be the fraction of the input target for grade J, category of enlistment C, and SSL I that supports (should be allocated to) lateral AFSC IL. It is calculated only for combinations I and IL such that I supports IL (there is a value of index such that I = FAF(IL,INDEX)).[3] It is calculated as follows:

```
If C = 1 AND J > 3 THEN SUPFR(J,C,I,IL) = 0
ELSE IF C NE 1 AND J > 3 THEN SUPFR(J,C,I,IL) =  AUTHC(IL,J) × X(IL,INDEX)/
                                                  sum(LTARG(J,m,y,c,I)),
```

where the sum is over all YETS, YOS, and c = 2, 3, and 4.

```
IF J = 3 THEN SUPFR(J,C,I,IL) = AUTHC(IL,J) × X(IL,INDEX)/
                                 sum(LTARG(J,m,y,c,I)),
```

where the sum is over all YETS, YOS, and c.

The next step is to calculate the array TTARG such that TTARG(J,YETS,YOS,C,IL,I) is the contribution of the indicated cells in the target for SSL I to the target for lateral IL. Again this is calculated only for combinations I and IL such that I supports IL:

[3] The prototype is implemented in SAS, which handles sparse matrixes without wasting space or calculation efforts.

TTARG(J,YETS,YOS,C,IL,I) = SUPFR(J,C,I,IL) × LTARG(J,YETS,YOS,C,I).

The target for the lateral is then obtained as the sum of its parts:

TARGET(J,YETS,YOS,C,IL) = sum(TTARG(J,YETS,YOS,C,IL,FAF(IL,index))),

where the sum is over all values of index.

The portion of the SSL target that is part of the target for a lateral is not available for the basic AFSC. Thus for any basic AFSC I,

TARGET(J,YETS,YOS,C,I) = LTARG(J,YETS,YOS,C,I) –
sum(TTARG(J,YETS,YOS,C,IL,I)),

where the sum is over all lateral AFSCs IL that are supported by I.

Note that this division of the target between lateral AFSCs and basic AFSCs means that the target in each grade for each lateral AFSC sums to its authorizations in that grade. Any shortfall between the grade target for the SSL and the authorizations for the SSL are borne entirely by the basic AFSC. The priority given to the lateral target in this procedure is consistent with other current Air Force procedu‑es (e.g., the IPM). If it becomes desirable to have a more balanced allocation of overages and shortages, then another step could be added to the calculation of the array SUPFR.

CAPPERS

The only remaining step removes the capper portions of the array TARGET from AFSCs that are capped and then constructs the target for each capper by summing the parts that were removed.

The process begins by initializing the variable CAP(I) for each AFSC I to the AFSC that caps I, if there is one, and to a missing value otherwise. This is the same variable that was constructed in Sec. II.

The process proceeds in the inverse order of the allocation of capper authorizations to their feeders: First it processes grade 6 cappers, then grade 8, then grade 9. At each level, the target allocated to the capper AFSC includes its share of the target for any AFSCs that cap the capper.

Step 1: Do for each AFSC I such that CAP(I) is an E-6 capper, and for each YOS, YETS, C, and for J = 6,7,8,9:

TARGET(J,YOS,YETS,C,CAP(I)) =TARGET(J,YOS,YETS,C,CAP(I)) +
TARGET(J,YOS,YETS,C,I)

TARGET(J,YOS,YETS,C,I) = 0.

Step 2: Do for each AFSC I such that CAP(I) is an E-8 capper, and for each YOS, YETS, C, and for J = 8,9:

TARGET(J,YOS,YETS,C,CAP(I)) = TARGET(J,YOS,YETS,C,CAP(I)) +
TARGET(J,YOS,YETS,C,I)

TARGET(J,YOS,YETS,C,I) = 0.

Step 3: Do for each AFSC I such that CAP(I) is an E-9 capper, and for each YOS, YETS, C, and for J = 9:

TARGET(J,YOS,YETS,C,CAP(I)) = TARGET(J,YOS,YETS,C,CAP(I)) +
TARGET(J,YOS,YETS,C,I)

TARGET(J,YOS,YETS,C,I) = 0.

At the conclusion of this step the array TARGET contains a detailed target for each AFSC. It can then be summarized as needed for the policy purpose. For example, the bonus effects model needs a target only by YOS and AFSC. This is constructed by summing TARGET(J,YOS,YETS,C,I) across all values of J, YETS, and C.

OUTPUT

The output from this module is the array TARGET, which has the same dimensions as the IPM's inventory. The BEM reads this file and displays the target by YOS. It is hoped that a target analysis module will be developed that will display additional information about the target. For the target analysis module to easily read targets for both AFSCs and for SSLs, it is desirable for the output file that contains the array target to be in the same form as the input file that contains the array LTARG.

REFERENCES

Carter, Grace M., *Year of Service Target Generator: Conceptual Specification,* RAND, N-3223/1-AF, 1991.

Carter, Grace M., et al., *Middle-Term Loss Prediction Models for the Air Force's Enlisted Force Management System: Specification and Estimation,* RAND, R-3482-AF, December 1987.

Carter, Grace M., Jan M. Chaiken, Michael P. Murray, and Warren E. Walker, *Conceptual Design of an Enlisted Force Management System for the Air Force,* RAND, N-2005-AF, August 1983.

Carter, Grace M., Deborah L. Skoller, Stanley E. Perrin, and Clyde S. Sakai, *An Enlisted Force Management System Model to Predict the Effects of Bonus Decisions,* RAND, N-2747-AF, July 1988.

Hadley, G., *Nonlinear and Dynamic Programming,* Addison-Wesley Publishing Company, Inc., Reading, Massachusetts, 1964.

Jaquette, D. L., G. R. Nelson, and R. J. Smith, *An Analytic Review of Personnel Models in the Department of Defense,* RAND, R-1920-ARPA, 1977.

Walker, Warren E., and the Enlisted Force Management Project Team, *Design and Development of an Enlisted Force Management System for the Air Force,* RAND, R-3600-AF, 1991.